AD-A085 486 AIR FORCE ENVIRONMENTAL TECHNICAL APPLICATIONS CENTER--ETC F/G 9/2
WIND FACTOR SIMULATION MODEL: USER'S MANUAL.(U)
APR 80 R C WHITON, P L HEROD
UNCLASSIFIED USAFETAC/TN-80/002 NL

1 OF 1
AD
A085486

END
DATE
FILMED
7 80
DTIC

LEVEL II

USAFETAC/TN-80/002

ADA085486

WIND FACTOR SIMULATION MODEL

User's Manual

Roger C. Whiton, Maj, USAF
Patrick L. Herod, Capt, USAF

April 1980

UNITED STATES AIR FORCE
AIR WEATHER SERVICE (MAC)

USAF ENVIRONMENTAL
TECHNICAL APPLICATIONS CENTER

SCOTT AIR FORCE BASE, ILLINOIS 62225

REVIEW AND APPROVAL STATEMENT

USAFETAC/TN-80/002, Wind Factor Simulation Model: User's Manual, April 1980, is approved for public release. There is no objection to unlimited distribution of this report to the public at large, or by the Defense Technical Information Center (DTIC) to the National Technical Information Service (NTIS).

This technical note has been reviewed and is approved for publication.

GERALD J. DITTBERNER, Maj, USAF
Chief, Aerospace Sciences Branch
Reviewing Officer

FOR THE COMMANDER

WALTER S. BURGMANN
Scientific and Technical
Information Officer (STINFO)

12 MAY 1980

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

14

1. REPORT NUMBER: USAFETAC/TN-8∅/∅∅2

2. GOVT ACCESSION NO.: AD-A085486

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle):

6 WIND FACTOR SIMULATION MODEL: User's Manual.

5. TYPE OF REPORT & PERIOD COVERED: 9 Final Report,

6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s):

10 Roger C./Whiton Maj, USAF
Patrick L./Herod Capt, USAF

8. CONTRACT OR GRANT NUMBER(s):

9. PERFORMING ORGANIZATION NAME AND ADDRESS:

USAF Environmental Technical Applications Center
Scott AFB, Illinois 62225

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:

11. CONTROLLING OFFICE NAME AND ADDRESS:

USAF Environmental Technical Applications Center
Scott AFB, Illinois 62225

12. REPORT DATE: 11 Apr[redacted]80

13. NUMBER OF PAGES: 66

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office):

12 66

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer applications; computer programs; computer program documentation; computerized simulation; equivalent headwind technique; great circle; great circle distance; great circle equation; great circle heading; great circle navigation; manual; meteorology; models; navigation; simulation; subroutines; user needs; weather; wind; wind factor; wind factor simulation.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

User instructions and a concise description are provided for a Wind Factor Simulation Model (WFSM). The WFSM is a fast, economical module designed to reside as a collection of subroutines within the user's larger simulation model. The WFSM, upon call by the user, produces mean overall climatological wind factors for great circle routes between arbitrary points "A" and "B" (specified by latitude and longitude) anywhere on the globe. The WFSM produces wind factors in any of three modes (calm wind case, 90-percent worst case, and the mean wind case), for either of two altitudes (25,000 ft and 35,000 ft) for any (Cont'd)

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

400945 Du

20. ABSTRACT (Cont'd)

of four seasons of the year. In addition, the model can provide great circle distance between points 'A' and 'B.' From this information and known airspeed, the user can calculate ground speed and adjusted flying time between 'A' and 'B.' Software solves the equation of a great circle. Program listing and flow chart are included.

USER'S MANUAL

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION 1.0 GENERAL

1.1 Purpose of the User's Manual. The objective of the user's manual (UM) for the Wind Factor Simulation Model (WFSM), USAFETAC Project 1923, is to provide the information necessary to use the model effectively. This user's manual is supplemented by USAFETAC/TN-80/001, Wind Factor Simulation Model, April 1980, which provides an in-depth treatment of the model's science and mathematics. This manual refers to USAFETAC/TN-80/001 frequently. Readers may wish to have the technical note available as supplementary reading.

1.2 Project References. The Wind Factor Simulation Model (WFSM) is embodied in a constellation of computer subroutines that calculate simulated wind factors for a larger simulation model in which the WFSM resides. Originally, the WFSM was written to serve the Military Airlift Command's (MAC) airlift system simulation called COLOSSUS. COLOSSUS is MAC's attempt to test by means of computer simulation its ability to respond to a contingency at any place or any time. MAC perceived a need to enhance the realism of COLOSSUS by adding simulated weather. The formal request for this weather support is dated 24 July 1978. This request called for three weather modules: (a) terminal weather for takeoff/departure; (b) enroute refueling weather; and (c) enroute wind factors. Efforts to meet requirement (a) began as early as 1977. This UM documents software developed to meet requirement (c).

1.2.1 Project Request. 7WW/DON letter to USAFETAC/DO, Request for Weather Data, 24 Jul 1978.

1.2.2 Documentation on the Project. USAFETAC/TN-80/001, Wind Factor Simulation Model: Model Description, April 1980.

1.2.3 Documentation Concerning Related Projects. None.

1.2.4 Standards or Reference Documentation.

1.2.4.1 Documentation Standards and Specifications. Documentation is in accordance with DOD Standard 7935.1-S, Automated Data Systems Documentation Standards, 13 September 1977, and AFM 171-100, 300-12, and 300-15.

1.2.4.2 Programming Conventions. American National Standard X3.9-1966 FORTRAN programming convention has been adhered to except in the following cases:

a. Extensive use of the Honeywell 6000-Series FORTRAN Execution Error Monitor (FXEM) has been made to assist in debugging.

b. Continuation statements are handled with an ampersand in column 1 rather than a nonblank character in column 6. This permits running the WFSM in Honeywell TSS YFORTRAN as well as in CARDIN.

1.2.4.3 DOD or Federal Standards. Documentation provided in the UM is in accordance with DOD Standard 7935.1-S, 13 September 1977.

1.3 Terms and Abbreviations:

Model: A representation, description, or imitation of a system or physical process (such as the atmosphere) in another medium (such as a computer). A model is a simplified, generalized conceptualization of complex reality, usually based on a set of simplifying assumptions needed to obtain tractable solutions. The model is so constructed as to behave similarly to its prototype system or physical process in some sense considered critical to the problem at hand. The model abstracts or preserves suitably chosen "essential" properties of the system or process being modeled.

Simulation: A numerical technique by which systems or processes are modeled digitally in order to study the behavior of the process or system being simulated, usually as a function of time. Simulation has been described as attaining the essence without the reality. Its purpose is to permit drawing conclusions about the real-world system or process through use of the simulator as a tool for study. All simulations are models, but not all models are simulators.

Environmental Simulation: A selectively realistic synthesis of aerospace behavior consistent in space and time. A technique, often involving mathematical and probabilistic models, of describing or analyzing the environment of the effects of the environment on the system. An environmental simulation model can stand alone or can operate within a larger simulation model such as COLOSSUS.

COLOSSUS: Name given to MAC's attempt to simulate their airlift forces' capability to respond to a contingency anywhere at any time.

Wind Factor: The difference between the ground speed of an aircraft and its true airspeed.

Ninety Percent Worst: A wind factor whose value is exceeded 90 percent of the time, i.e., there is a 10-percent risk that one will encounter a wind factor more negative than the 90-percent worst wind factor.

Great Circle Distance (GCD): The distance between two points computed along a great circle of the earth. A great circle of the earth is any circle around the surface of the earth whose center coincides with the center of the earth. All longitude lines are great circles. The only latitude line that is a great circle is the equator.

Newton's Iterative Method: A mathematical technique applied in subroutine GRTCIR to solve a transcendental equation (see USAFETAC/TN-80/001).

Sawyer's Equivalent Headwind Technique: The mathematical technique used in the WFSM to compute wind factors. For details see USAFETAC/TN-80/001 and AWS-TR-77-267, Guide for Applied Climatology, November 1977.

1.4 Security and Privacy. This UM and the Wind Factor Simulation Model (WFSM) for which it is written are unclassified and can be released to the public. No privacy restrictions are associated with the use of this system to include input, output, data base or programs.

SECTION 2.0 SYSTEM SUMMARY

2.1 System Application. The Wind Factor Simulation Model (WFSM) was designed and developed to be incorporated in the Military Airlift Command's simulation of airlift forces. This advanced computer simulation effort is called COLOSSUS.

Designers of the COLOSSUS simulation determined that three weather elements most seriously impact airlift system operations: terminal weather for departure/recovery, enroute visibility for inflight refueling, and enroute winds. The WFSM was developed in response to the third COLOSSUS requirement. Wind factors generated by the WFSM enhance the effectiveness of the COLOSSUS simulation by enabling calculation of realistic flight times based on climatological winds.

The WFSM provides to COLOSSUS the ground speed of an aircraft on a simulated flight from point "A" (takeoff) to point "B" (landing), two points anywhere on the globe. It does this by calculating an overall enroute mean wind factor between these points by the Sawyer equivalent headwind technique and then applying the result to a simple algebraic expression for ground speed. For details on how this is accomplished, see USAFETAC/TN-80/001.

There are at least two other applications for the WFSM. First, it can be used, with appropriate modifications, as an efficient method of generating wind and great circle navigation information. It is efficient in terms of computer run time and core storage. Its efficiency is especially evident in comparisons between the current operational wind factor models and the WFSM. In addition, this module, again with required changes, can be used in other simulators. Simulation efforts are increasing in number and complexity. It follows that a requirement for a module incorporating weather information should be included in these simulation efforts. The WFSM can be a partial fulfillment of that requirement.

2.2 System Operation. Before the WFSM is first invoked, a user-supplied initializing module must read the wind data base from a sequential file (card, tape, or disk; but as implemented on MAC's Honeywell Series-6000 computer, a disk file) into the computer's memory.

After the data base has been loaded into the computer's memory, the user at any time may invoke the WFSM by calling its main subroutine, ENRWND. In that call, the user tells ENRWND the location of point "A" (takeoff) and point "B" (landing), the aircraft altitude and airspeed, and the date/time for which the wind factor is requested. The WFSM then responds by stepping the aircraft along a great circle route between "A" and "B", calculates a route-mean wind factor, adds that wind factor to the given airspeed, and returns route-mean ground speed to the user. The process is illustrated in Figure 1.

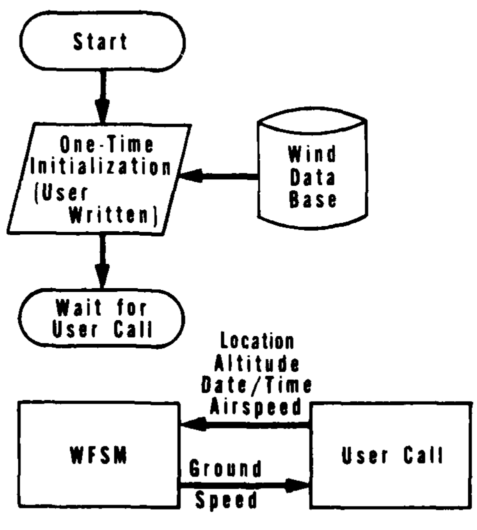


FIGURE 1. System Operation.

2.3 System Configuration. The WFSM was developed for use on the MAC Honeywell 6000-Series computers at Scott AFB, Illinois. In particular, the WFSM executes on a Honeywell 6080 computer. During program design and coding, the goal of intercomputer compatibility was for the most part adhered to. With exceptions noted in paragraph 1.2.4.2, the WFSM should compile and execute on any computer with a FORTRAN compiler.

The wind data base for the WFSM resides in the computer's core storage during execution and on a sequential file (card, tape, or disk) before execution. A user-supplied initializing module external to the WFSM must read that data base from the file into the computer's memory before WFSM is first invoked. As implemented on the MAC Honeywell 6080, the wind data base resides on a permanent sequential disk file.

Configuration necessary for a computer to use WFSM typically requires a FORTRAN compiler, linking loader, disk storage, and FORTRAN disk input/output capability.

2.4 System Organization. The WFSM is coded as a constellation of seven FORTRAN subprograms with the functions described in Table 1. The principal subroutine is ENRWND, with which the user's program communicates directly via arguments of the FORTRAN CALL statement. Subroutine ENRWND, in turn, calls other subprograms, some of which call still other subprograms. The hierarchy of subroutine calls is shown in Figure 2.

TABLE 1. WFSM SUBPROGRAMS.

Subprogram	Purpose
ENRWND	Main enroute wind factor subprogram, called directly by user to compute ground speed from given airspeed in any of three modes for two flight levels and four seasons of the year.
DISTAN	Called by ENRWND and HDG or directly by user to compute great circle distance between any two points on the globe.
SPHGLO	Conversion of spherical coordinates to global latitude and longitude or vice-versa. Called by GRTCIR.
GRTCIR	Solution of great circle equation for latitude given longitude, or for longitude given latitude. Called by ENRWND.
HDG	Calculation of initial heading along a great circle route flown from a given origin to a given destination. Called by ENRWND and GRTCIR.
BRLNG	For a spherical grid system whose longitude boundaries are spaced at 30° intervals, finds the longitude grid values bracketing a given arbitrary longitude. Called by ENRWND.
BRLAT	For a spherical grid system whose latitude boundaries are spaced at 15° intervals, finds the latitude grid values bracketing a given arbitrary latitude. Called by ENRWND.

Detailed explanation of the processing done by each subprogram is available in USAFETAC/TN-80/001.

As shown in Figure 3, the WFSM is designed to operate as a miniature, subservient simulation within the user's overall simulation model. As such, WFSM is supplied as

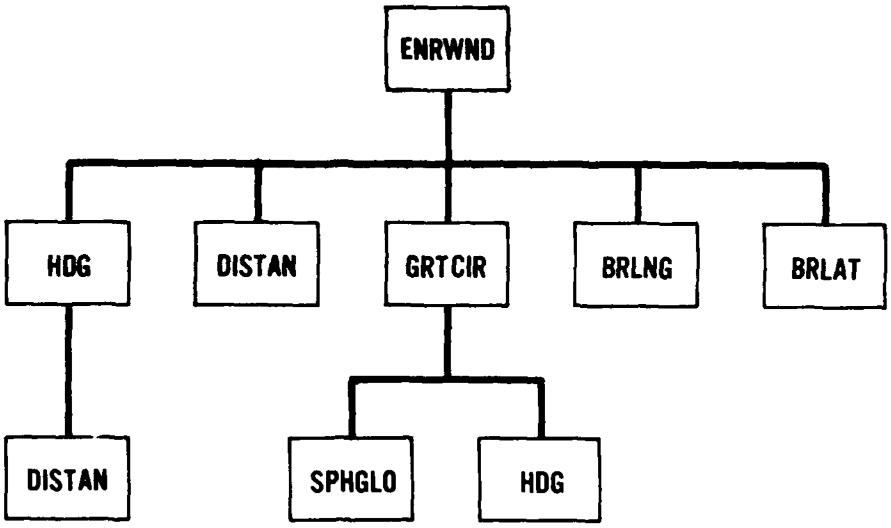


FIGURE 2. Hierarchy of Subroutine Calls.

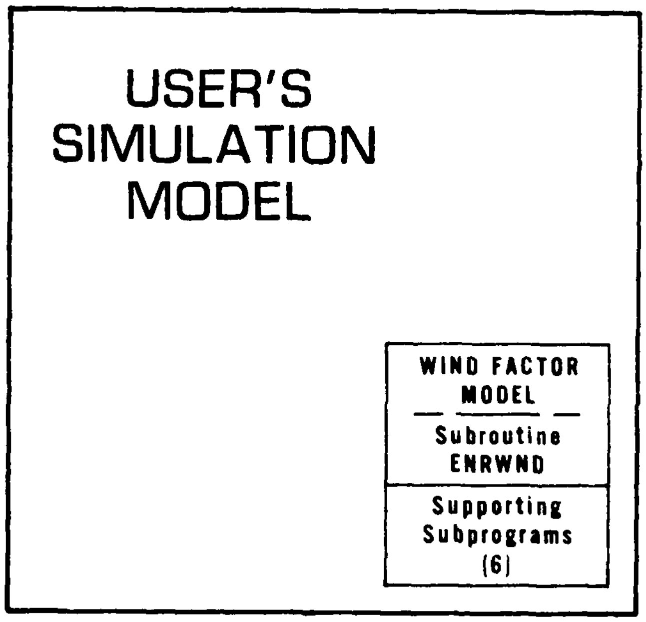


FIGURE 3. WFSM as Server-Model within User's Simulation Model.

a constellation of FORTRAN subroutines. Even the main WFSM module, called ENRWND, is a subroutine. All communication between WFSM and the encompassing overall simulation is done through the arguments of the user's CALL statement and corresponding arguments of the SUBROUTINE ENRWND statement.

2.5 Performance.

2.5.1 Capabilities. The WFSM is capable of producing mean overall climatological wind factors for great circle routes. It does so in any of three modes: calm wind case, 90-percent worst wind case, and the mean wind case. The wind factor can be produced for two flight levels and for four seasons. In addition, the model, through its subroutine DISTAN, can provide the great circle distance (GCD) between any two points over the globe. Since it produces both wind factors and GCD, it is capable of calculating both the ground speed and the adjusted flying time between points "A" and "B" if given the airspeed.

2.5.2 Assumptions. The WFSM assumes that all routes are great circle routes or can be subdivided into several legs each of which is a great circle segment. It is further assumed that climb winds and descent winds play a negligible role in determining the route-mean wind factor. In the model as presently written, the simulated aircraft is always "at altitude." Since an arithmetic rather than a harmonic mean is used in computing the route-mean wind factor, it is assumed that the ground speed is less than or equal to one third of the airspeed.

2.5.3 Limitations. Presently, the WFSM cannot produce a "simulated" wind having a day-to-day variability. Hence, within any given season and for a particular altitude, a particular route will always experience the same wind factor, regardless of the passage of time. Requests for the simulated wind default to the mean wind.

Furthermore, the present model is incapable of producing a "forecast" wind factor. Requests for a forecast wind default to the mean wind.

Temperature, aircraft performance, fuel consumption and other elements considered by typical flight planning models are not included in the present simulation, which deals only with wind.

Operations north of 75°N or south of 60°S are not permitted.

The coarseness of the present 15° latitude by 30° longitude grid precludes use of the model for calculation of the operationally realistic wind factors unless changes are made. Simple modifications to the existing data base and grid system, along with corresponding changes in the software, can remove this limitation.

Specifically, the following restrictions must be adhered to:

a. No route should be flown directly over either pole.

b. Neither pole should serve as a point "A" or "B."

c. Circumferential or round-robin flights in which "A" and "B" coincide will be aborted. Such flights should be broken into smaller segments.

d. Routes or segments of routes flown directly north or directly south along a longitude line will also abort.

e. At present, the model allows flights only at altitudes of 25,000 feet and 35,000 feet.

f. All departure and destination points must lie within the latitude range from 75°N to 60°S.

Restrictions e and f were imposed in order to reduce the core storage requirements of the model.

2.5.4 Processing Time. Generating a wind factor for a simulated aircraft flying a long route (4,725 nautical miles) requires approximately 0.1-0.5 seconds of central processing unit (CPU) time on a Honeywell 6080 general purpose computer. The CPU time depends on the number of legs in the simulated flight. Accordingly, this CPU time, referring as it does to a very long route, represents almost a worst case timing estimate.

2.5.5 Flexibility. A number of the limitations of the WFSM can be overcome by selective modification of the model, the data base, or both. Since the WFSM was built for economy in run time and storage, such modifications will have the side effect of increasing run time and/or storage. In particular, the resolution of the WFSM grid can easily be improved by changing delta-latitudes and delta-longitudes in the model, altering the grid numbering scheme, and adding to the data base. In addition, the number of vertical levels and time periods can be increased. With more effort, WFSM capabilities can be expanded to include a "simulated" (i.e., variable) wind by writing Part V of subroutine ENRWND, which is presently set to default to the mean wind.

2.5.6 Error Detection. User inputs to the WFSM are scanned for legal values. In iterative calculations, iteration counters are used to flag computations that are diverging or failing to converge. The Honeywell 6000-Series FORTRAN Execution Error Monitor (FXEM) is invoked to display to the user any errors so identified. The FXEM error message, including error number and a plain text diagnostic, are displayed on SYSOUT. Error number 61 has been used for all WFSM errors. Whenever FXEM error 61 appears, the user knows that a problem associated with the WFSM has been detected. Each FXEM error has a disposition. Some permit the program to continue executing, while others produce a program abort. Presently, FXEM error 61 causes program abort. Users can alter this disposition by means of initializing call to the FXEM module. Errors and problems detected and treated by the WFSM are summarized in Table 2.

2.6 Data Base and Grid System

2.6.1 Data Base. Basic data for the WSFM are the USAFETAC Single Integrated Operational Plan (SIOP) winds. With a period of record extending from January 1972 to December 1976, the SIOP winds contain mean u-component, v-component, and vector standard deviation, tabulated on a 5-degree offset grid. This grid is much finer than that used by the WFSM.

To prepare a data base for the WFSM's 15° latitude by 30° longitude grid system, simple averaging is used. The u-component, the v-component, and the vector standard deviation are averaged separately. The average is performed by extracting from the SIOP winds three values latitudinally and six values longitudinally. The resulting 18 values are summed and the sum divided by 18 to obtain the mean data values for each grid sector. In this way, mean u-component, v-component, and vector standard deviation are obtained for all 108 grid sectors. Such a data base is constructed for January (winter), April (spring), July (summer), and October (fall) for altitudes 25,000 feet (taken from 400-mb winds) and 35,000 feet (taken from 250-mb winds). Using an in-house USAFETAC program WIND, the

$$4 \text{ season} \times 2 \text{ altitude} = 8 \text{ sets}$$

of 108-sector wind data are converted from u, v, and vector standard deviation to direction (beta angle, described in Appendix B, USAFETAC/TN-80/001), speed, and variance. After conversion, the winds are stored sequentially in a data file in the order described in Table 3. Each sector of the data file requires one record of disk storage containing the three elements of information shown in Table 4.

There are

$$4 \text{ seasons} \times 2 \text{ altitudes} \times 108 \text{ sectors} = 864 \text{ records}$$

of wind data in the data file. To be used by the WFSM, the data base must be loaded from the data file into the arrays,

DIR(870)	Wind Direction, Beta Angle, Radians
SPD(870)	Wind Speed, Knots
VAR(870)	Wind Variance, Knots^2

which are located in the COMMON block WEA and require 2.6K words of computer core storage. The COMMON block must be loaded with wind data before the WFSM is first executed. Most often, an initializing routine external to the WFSM is used for this purpose. That routine merely reads the 864 wind records in order and stores them,

TABLE 2. ERROR DETECTION FEATURES OF WFSM.

Subroutine	Error Description/MESSAGE	FXEM Display	Disposition
ENRWND	Wind option less than 0 or greater than 3 ILLEGAL WIND OPTION	Yes	Abort
ENRWND	Latitude or longitude out-side legal range ILLEGAL LAT/LON	Yes	Abort
ENRWND	Julian date greater than 366 or less than 1 ILLEGAL JULIAN DATE	Yes	Abort
ENRWND	Altitude index less than 1 or greater than 2 ILLEGAL ALTITUDE	Yes	Abort
ENRWND	Subroutine GRTCIR twice consecutively failed to converge when solving for longitude of Candidate Point #2 GRTCIR FAILED TWICE	Yes	Abort
ENRWND	Number of legs of the simu-lated flight exceeds a variable maximum number currently set to 50 RUNAWAY ROUTE	Yes	Abort
GRTCIR	Subroutine function vector less than 1 or greater than 5 ILLEGAL ICONV	Yes	Abort
GRTCIR	Direct north-south flight is attempted SOLUTION NOT UNIQUE IN THETA	Yes	Abort
GRTCIR	Newton's iterative method fails to converge, as indi-cated by an iteration count exceeding a variable maximum currently set at 7. Because of the way in which GRTCIR is used by ENRWND, failure to converge is sometimes normal behavior. In these cases no error message is displayed and the program continues. When, the GRTCIR function flag, ICONV, is set to 2, an error message SOLUTION DID NOT CONVERGE is displayed, and the pro-gram aborts.	No Yes	Continue Abort
SPHGLO	Erroneous function flag ICONV. Error code ICONV=10 returned	No	Continue
SPHGLO	Out-of-bounds latitude. Error code ICONV=11 returned	No	Continue

SPHGLO	Out-of-bounds longitude. Error code ICONV=12 returned	No	Continue
SPHGLO	Out-of-bounds θ Error code ICONV=13 returned	No	Continue
SPHGLO	Out-of-bounds φ Error code ICONV=14 returned	No	Continue

Table 3. STRUCTURE OF WIND DATA FILE.

Season	Flight Level	Record Numbers
Winter	250	1 - 108
	350	109 - 216
Spring	250	217 - 324
	350	325 - 432
Summer	250	433 - 540
	350	541 - 648
Fall	250	649 - 756
	350	757 - 864

Table 4. WIND DATA FILE RECORD ELEMENTS.

Element	Units	Columns	FORTRAN Format
Wind Direction (Beta Angle)	radians	1-15	F15.5
Wind Speed	knots	16-30	F15.5
Variance	$knots^2$	31-45	F15.5

without rearrangement, in the COMMON block WEA. Thereafter, the wind data base can be accessed in terms of a record number NRND as follows:

DIR(NRND) SPD(NRND) VAR(NRND)

The record number NRND is computed as follows:

NRND = (Season Index - 1) x (Total Number Altitudes) x
(Total Number Grid Sectors)
+ (Altitude Index - 1) x (Total Number Grid Sectors)
+ (Sector Number)

2.6.2 Grid System. The WFSM is required to be economical in terms of computer core storage and run time; yet the model must provide a global wind factor capability to the user. These conflicting requirements dictated the use of a coarse grid. The original idea, motivated by the latitude/longitude orientation of the winds used as input data, was to have a grid system in global (latitude/longitude) coordinates. The grid would have a spacing of 15° latitude by 30° longitude and would cover the globe, requiring 144 grid sectors (12 sectors from pole to pole times 12 sectors around the equator). Closer investigation of the latitudes of the COLOSSUS terminals and routes indicated that grid sectors above 75°N and below 60°S were unnecessary. Since three quantities (wind direction, speed, and variance) are stored for each grid sector for four seasons and two flight levels, approximately 0.87K words of core storage could be saved by reducing the number of grid sectors from 144 to 108.

The grid system chosen for implementation is in global (latitude/longitude) coordinates with a resolution of 15° latitude by 30° longitude. The grid system extends from 75°N to 60°S latitude (9 rows) and globally in longitude (12 columns). Hence, there are 108 grid sectors. Numbering of the grid sectors is shown in Figure 4. It is important to note that column 1 is keyed on 30°W, not 0°.

Column # →

Row # ↓	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2	13	14	15	16	17	18	19	20	21	22	23	24
3	25	26	27	28	29	30	31	32	33	34	35	36
4	37	38	39	40	41	42	43	44	45	46	47	48
5	49	50	51	52	53	54	55	56	57	58	59	60
6	61	62	63	64	65	66	67	68	69	70	71	72
7	73	74	75	76	77	78	79	80	81	82	83	84
8	85	86	87	88	89	90	91	92	93	94	95	95
9	97	98	99	100	101	102	103	104	105	106	107	108

75°N 60°N 30°N 0° 30°S 60°S

0° 90°E 180° 90°W

FIGURE 4. Grid System for WFSM.

A grid system such as this, involving undefined regions near the poles, prevents flights over the poles (which would be complicated by the singularity in coordinate systems at the poles). The coarseness of the present grid system precludes use of this model for calculation of operational wind factors. A coarse grid is suitable for simulation, however. The grid could be made finer if improved meteorological resolution is desired.

2.7 General Descriptions of Inputs, Processing, Outputs.

2.7.1 Input. A user employing the WFSM to calculate a route-mean wind factor and ground speed must stipulate the global (latitude/longitude) coordinates of point "A" (takeoff) and point "B" (landing), the Julian base date of the wind factor request (used to determine season of the year), aircraft altitude (25,000 ft or 35,000 ft), wind option (calm, mean, or 90-percent worst), and airspeed.

To accommodate future growth, the user must also stipulate Greenwich mean base time of the wind factor request and forecast hours ahead of the wind factor requested. Because the WFSM does not as yet have a capability to generate forecast wind factors, these two inputs are ignored and may have any value.

2.7.2 Processing. The keystone subprogram of the WFSM is ENRWND, which accomplishes most of the aircraft navigation except solution of the equation of a great circle. It also accomplishes all of the meteorology involved in calculation of the wind factor (see Chapter 3, USAFETAC/TN-80/001).

The ENRWND subprogram navigates a simulated aircraft over a great circle route from point "A" to point "B" in a global (latitude/longitude-oriented) grid system whose resolution is 15° latitude by 30° longitude. The subroutine determines which grid sector the aircraft is in and the length of the aircraft's path through the sector. Then the routine consults the wind data base for the particular grid sector, flight level, and season to obtain sector-mean wind direction, speed, and variance. Cross-track and along-track components of the wind are computed. These are weighted by the length of the aircraft's track through the sector and are accumulated. At the end of the simulated flight, the accumulators are divided by the total great circle distance for the flight, producing a distance-weighted, route-mean cross-track wind component, an along-track component and a variance. From these a route-mean wind factor is computed using Sawyer's equivalent headwind technique (see Chapter 3, USAFETAC/TN-80/001). If the 90-percent worst wind option has been selected, that wind factor is statistically adjusted to the 10-percent risk value. The wind factor is then added to the airspeed to obtain the route-mean airspeed. If the calm wind option is requested, all these computations are bypassed, and a zero wind factor is used. Presently, requests for the simulated wind option will default to the mean wind. In the mean wind case, the wind factor is developed directly from a mean wind data base. In the 90-percent worst case, this mean wind factor is statistically adjusted to a value such that 10 percent of the flights over that route will experience a worse wind factor (10-percent risk).

2.7.3 Outputs. The output from the WFSM (subroutine ENRWND) is the variable GSPEED, a real quantity representing the aircraft's ground speed in knots.

SECTION 3.0 STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

3.1 Initiation Procedures. The Wind Factor Simulation Model (WFSM) operates within a larger model, such as MAC's COLOSSUS. The larger model invokes the WFSM by issuing calls to its subroutines, principally subroutine ENRWND. Before the first call to WFSM, the WFSM must have been initialized by reading wind data from the data base into the COMMON block WEA. Users must provide this special initializing routine, as the WFSM code assumes wind data have already been read into core. A suitable routine is shown below:

```
        COMMON /WEA/ DIR(870), SPD(870), VAR(870)

        ...

        DO jjj NREC = 1,864

        READ (fc,iii) DIR(NREC), SPD(NREC), VAR(NREC)

    iii FORMAT (3F15.5)

    jjj CONTINUE

        ...
```

In this example, iii and jjj are integer FORTRAN statement numbers. The file code for the wind data base is fc.

Job control language (JCL) or its equivalent is needed to equate the logical file fc with a catalogued file name on the particular computer system used. Assuming that the wind data base is on a sequential disk file called PWIND.SQ within the subcatalogue MACRO in catalogue MACRO, then the Honeywell 6000-Series JCL is

```
$       PRMFL    fc,R,S,MACRO/MACRO/PWIND.SQ
```

where fc is the file code selected above. The $ PRMFL JCL must appear behind the $ EXECUTE record in a Honeywell job. Structure and format of the data base contained in the file PWIND.SQ is outlined in Table 3.

3.2 Staff Input Requirements. The WFSM communicates with the user's simulation model only through the argument list of the user's call to subroutine ENRWND.*

Whenever a wind factor is desired in the user's model, the user should place a call to ENRWND, supplying the values for input arguments. Subroutine ENRWND then takes control, calculates a route-mean wind factor and uses it to prepare the output argument, namely the ground speed of the aircraft. The user's call to subroutine ENRWND should be of the form,

```
CALL ENRWND (FRMLAT, FRMLNG, TOLAT, TOLNG, JULDAT,

 GMT, FCHRS, IALT, IOPTN, ASPEED, GSPEED)
```

Inputs to the WFSM are in the form of input arguments to subroutine ENRWND. These input arguments include the global (latitude/longitude) coordinates of point "A" (takeoff) and point "B" (landing). The user specifies which one of the four seasons he needs by stipulating a Julian base date (JULDAT) from 1 to 366. The user requests calm, mean, or 90-percent worst winds by specifying a wind option (IOPTN) from 0 to 3. The user then specifies the altitude of the aircraft by setting the flag IALT to 1 for 25,000 feet or 2 for 35,000 feet, and airspeed in knots in the input variable ASPEED.

*The exception is that user's may call subroutine DISTAN directly if they need a great circle distance, or function HDG if they want an initial great circle heading.

To accomodate future growth, the user must also stipulate GMT, the Greenwich mean base time (0.0 - 24.0) of the wind factor request, and FCHRS, a forecast hours ahead of the wind factor required. Presently, these inputs are ignored and may have any value.

A description of these input arguments is contained in Table 5.

TABLE 5. INPUT ARGUMENTS TO SUBROUTINE ENRWND.

Variable Name	Data Type	Definition	Units/Values
FRMLAT	Real	Latitude of point "A" (Takeoff)	Decimal degrees (75.0° to -60.0°)
FRMLNG	Real	Longitude of point "A" (Takeoff)	Decimal degrees (180.0° to -180.0°)
TOLAT	Real	Latitude of point "B" (Landing)	Decimal degrees (75.0° to -60.0°)
TOLNG	Real	Longitude of point "B" (Landing)	Decimal degrees (180.0° to -180.0°)
JULDAT	Integer	Julian base date of wind factor request	1 - 366
GMT	Real	Greenwich base time of wind factor request	0.0 - 24.0 hrs
FCHRS	Real	Forecast hours ahead	$\geq$ 0.0 hrs
IALT	Integer	Altitude index	1 = 25,000 ft 2 = 35,000 ft
IOPTN	Integer	Wind option or mode	0 = Calm wind 1 = Simulated wind (Defaults to 3) 2 = 90-percent worst wind 3 = Mean wind
ASPEED	Real	Airspeed	Knots

3.2.1 Input Formats and Composition Rules. Input arguments for subroutine ENRWND are presented in Table 5. Special format rules are as follows:

a. Latitudes and longitudes must be in decimal degrees, not the degrees-minutes-seconds system, e.g., 36.5° rather than 36° 30'.

b. South latitudes and east longitudes must be negative.

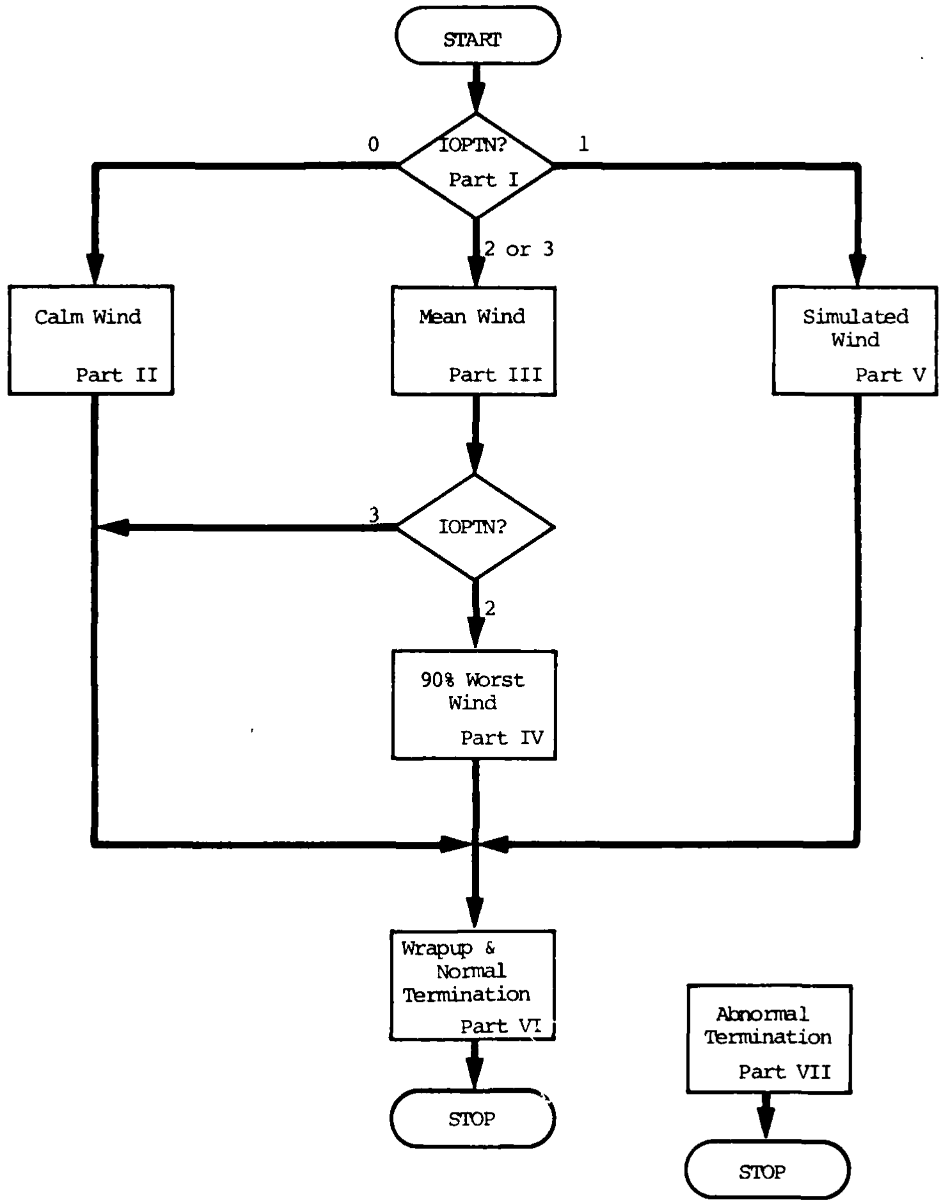
3.3 Output Requirements. Output argument for subroutine ENRWND is the variable GSPEED, a real quantity representing the aircraft's ground speed in knots. In addition, users may call subroutine DISTAN directly for a great circle distance, or function HDG for an initial great circle heading.

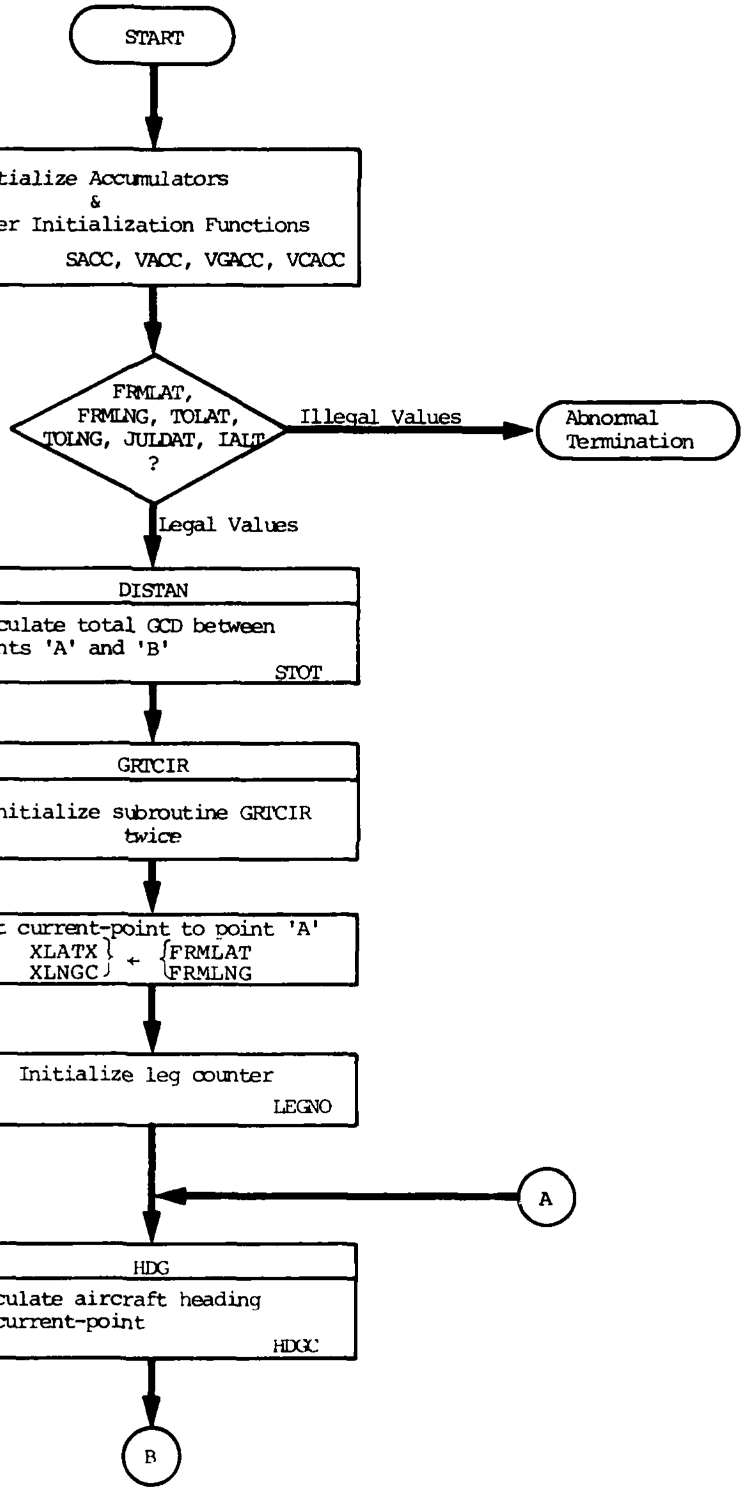
3.4 Utilization of System Outputs. The calling program, e.g., a simulation such as MAC's COLOSSUS, uses the WFSM to obtain the ground speed of an aircraft flying from point "A" to point "B". In order to provide the ground speed, the WFSM must calculate a wind factor. Most of the work in this module occurs in finding that wind factor. Once this is known, it is a simple matter to calculate the ground speed (see section 2.1 above and USAFETAC/TN-80/001). The ground speed information is important to the user's simulation because from it the adjusted flying time can be calculated. The user's simulation then has a timing factor to use for mission planning, war gaming, judging contingency response capabilities, and other applications.

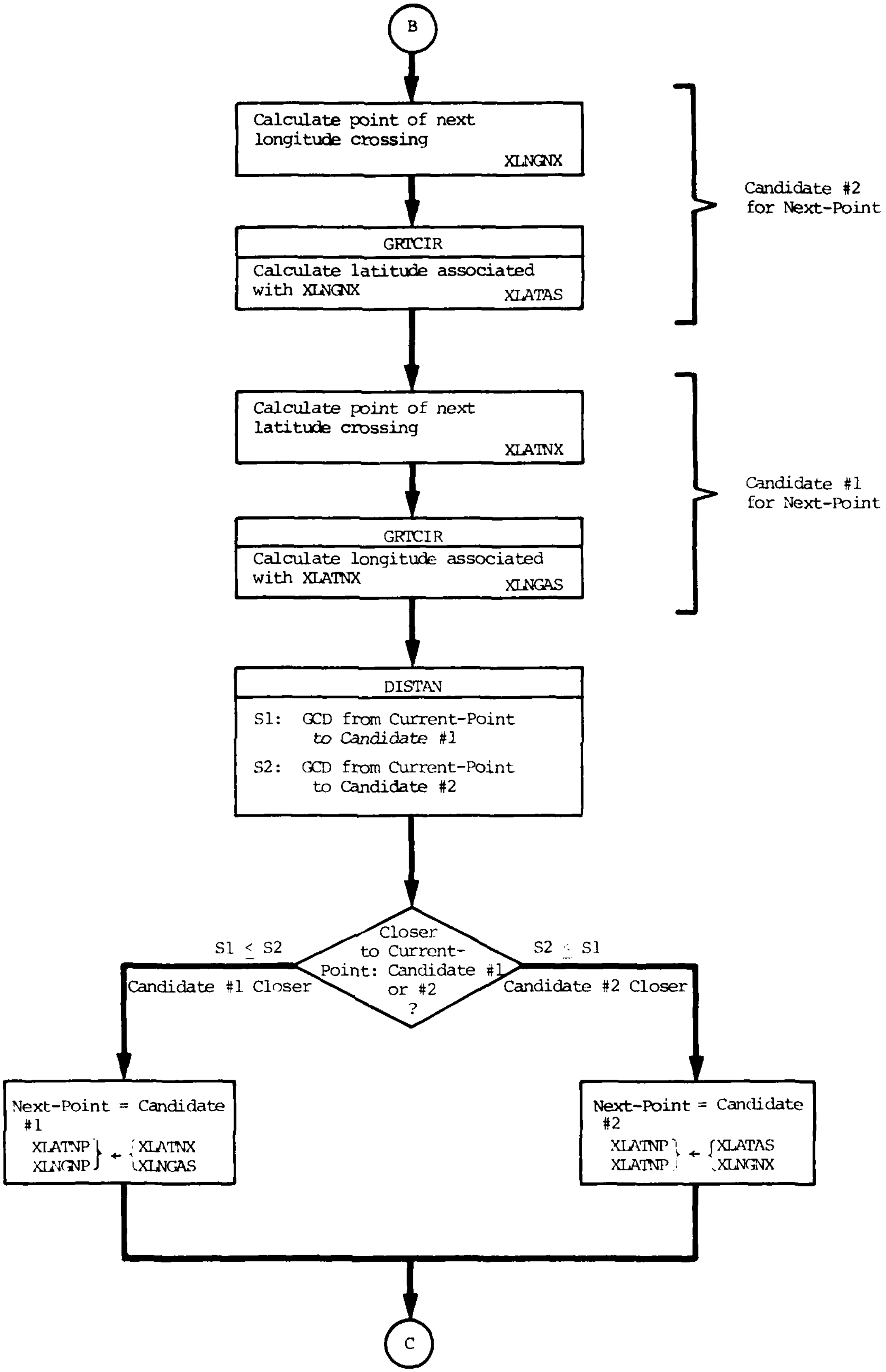
3.5 <u>Recovery and Error Correction Procedures.</u> To assist in debugging efforts, considerable use of the Honeywell Series 6000 FORTRAN Execution Error Monitor (FXEM) has been made. Whenever FXEM error 61 appears, there is a problem in the wind factor routines. A plain text diagnostic, indicating the nature of the problem, will appear on SYSOUT. Presently, FXEM error 61 is set to abort execution of the main program. By a call to one of the FXEM initializing routines, the user may set the disposition of FXEM error 61 to "continue" rather than "abort." No explicit restart features have been incorporated into the WFSM. Such procedures would not be feasible in an iterative, space-stepped solution.

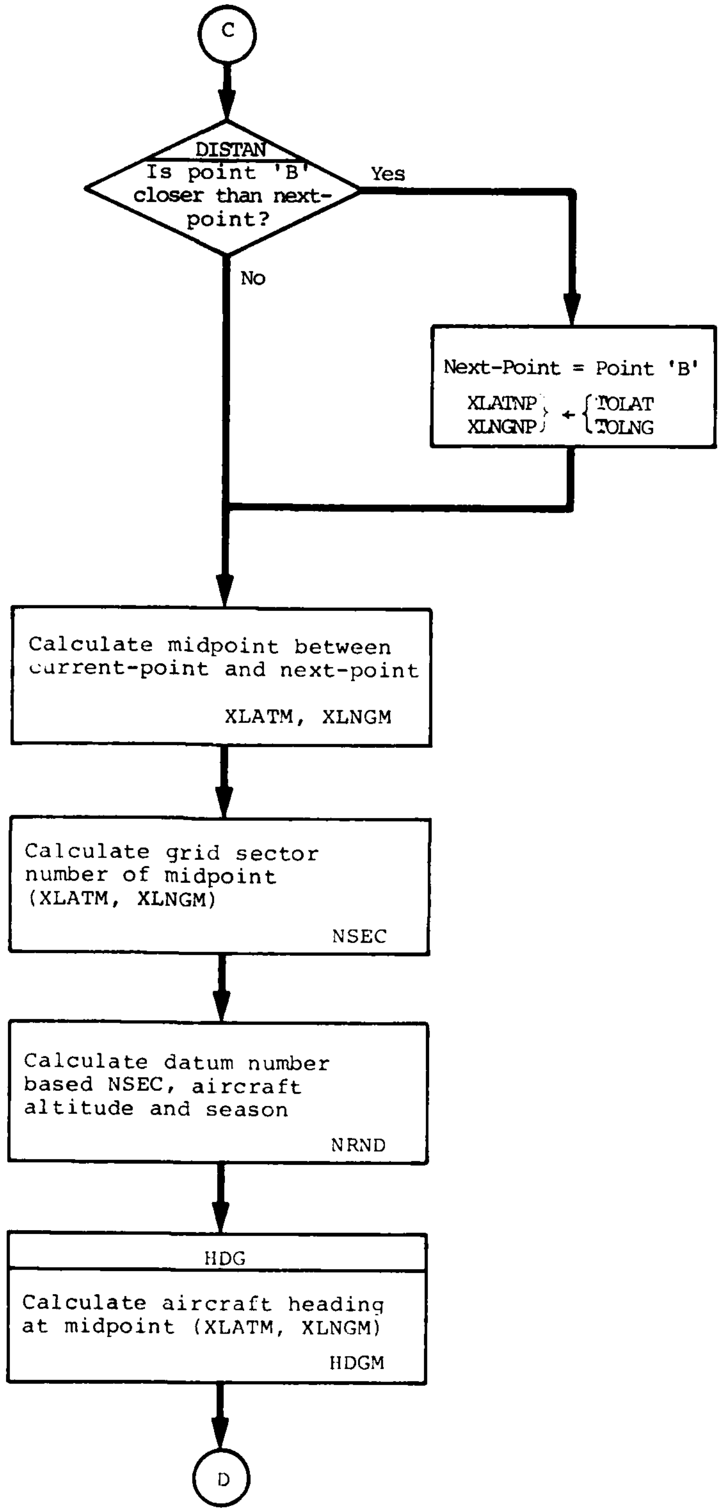
Appendix A

WFSM FLOW CHART




START
Initialize Accumulators
&
Other Initialization Functions
SACC, VACC, VGACC, VCACC
FRMLAT,
FRMLNG, TOLAT,
TOLNG, JULDAT, IALT
?
Illegal Values
Abnormal
Termination
Legal Values
DISTAN
Calculate total GCD between
Points 'A' and 'B'
STOT
GRTCIR
Initialize subroutine GRTCIR
twice
Set current-point to point 'A'
XLATX
XLNGC
FRMLAT
FRMLNG
Initialize leg counter
LEGNO
A
HDG
Calculate aircraft heading
at current-point
HDGC
B


B
Calculate point of next longitude crossing
XLNGNX
GRTCIR
Calculate latitude associated with XLNGNX
XLATAS
Candidate #2 for Next-Point
Calculate point of next latitude crossing
XLATNX
GRTCIR
Calculate longitude associated with XLATNX
XLNGAS
Candidate #1 for Next-Point
DISTAN
S1: GCD from Current-Point to Candidate #1
S2: GCD from Current-Point to Candidate #2
Closer to Current-Point: Candidate #1 or #2 ?
S1 ≤ S2
Candidate #1 Closer
Candidate #2 Closer
Next-Point = Candidate #1
XLATNP XLNGNP ← XLATNX XLNGAS
Next-Point = Candidate #2
XLATNP XLATNP ← XLATAS XLNGNX
C


C
DISTAN
Is point 'B' closer than next-point?
Yes
No
Next-Point = Point 'B'
XLATNP } ← { TOLAT
XLNGNP } ← { TOLNG
Calculate midpoint between current-point and next-point
XLATM, XLNGM
Calculate grid sector number of midpoint (XLATM, XLNGM)
NSEC
Calculate datum number based NSEC, aircraft altitude and season
NRND
HDG
Calculate aircraft heading at midpoint (XLATM, XLNGM)
HDGM
D

D

Convert heading at midpoint (XLATM, XLNGM) to angle α

ALPHA

↓

Get direction DIRM, speed SPDM, and variance VARM of wind at datum number NRND

↓

Compute along-ground-track wind component at midpoint

$$VGM = SPD \cdot \cos(|DIRM-ALPHA|)$$

↓

Compute cross-ground-track wind component at midpoint

$$VCM = SPDM \cdot \sin(DIRM-ALPHA)$$

↓

Compute time-averaged wind factor for present leg of route

$$WGAR = VGM - \frac{\left(VCM^2 + \frac{VARM}{2}\right)}{2 \cdot ASPEED}$$

↓

Accumulate distance-weighted components to accumulators

SACC: distance
VACC: variance
VGACC: along-track component
VCACC: cross-track component

↓

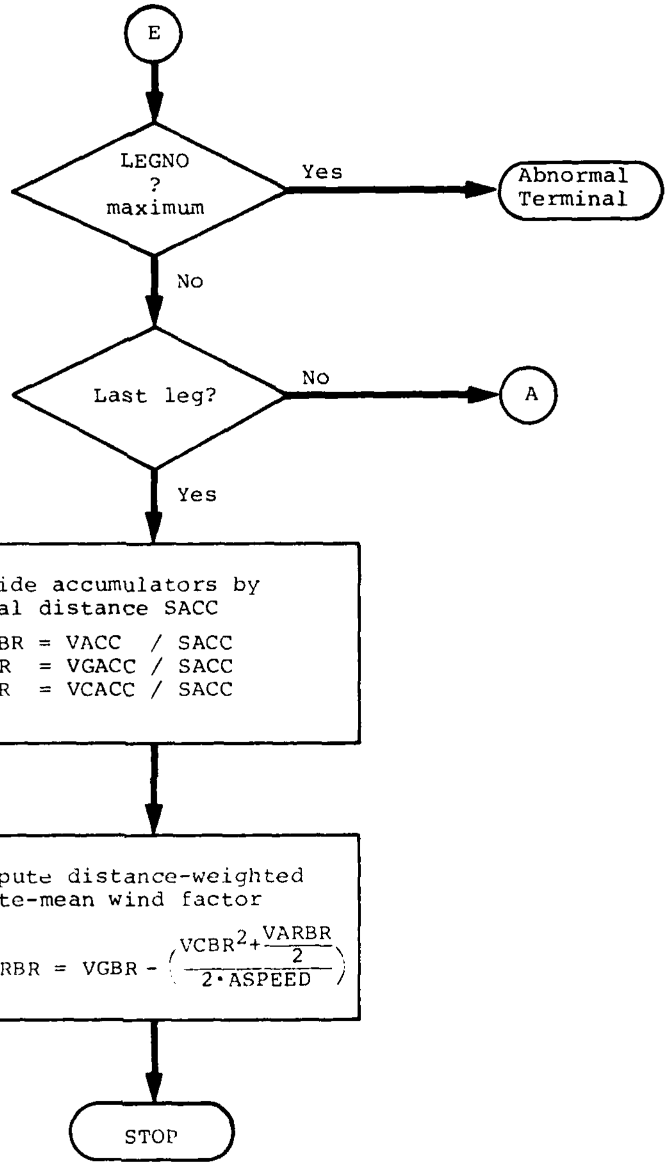
Advance to next-point.
Current-point ← next-point.
Increment leg number.

$$\begin{Bmatrix} XLATC \\ XLNGC \end{Bmatrix} \leftarrow \begin{Bmatrix} XLATNP \\ XLNGNP \end{Bmatrix}$$

$$LEGNO = LEGNO + 1$$

↓

E


E
LEGNO
?
maximum
Yes
Abnormal Terminal
No
Last leg?
No
A
Yes
Divide accumulators by total distance SACC
VARBR = VACC / SACC
VGBR = VGACC / SACC
VCBR = VCACC / SACC
Compute distance-weighted route-mean wind factor
$WGARBR = VGBR - \left(\frac{VCBR^2 + \frac{VARBR}{2}}{2 \cdot ASPEED} \right)$
STOP

Appendix B

WFSM PROGRAM LISTING

```
CENRWND ENROUTE WIND/R. C. WHITON/27 FEB 1979
C
      SUBROUTINE ENRWND (FRMLAT, FRMLNG, TOLAT, TOLNG, JULDAT, GMT,
&       FCHRS, IALT, IOPTN, ASPEED, GSPEED)
C
C************************************************************************
C*
C*    PROGRAM ID-     ENRWND
C*    MET ANALYST-    MAJ ROGER C. WHITON, USAFETAC/DNS, EXT 5412
C*    SYS ANALYST-    MAJ ROGER C. WHITON, USAFETAC/DNS, EXT 5412
C*    PROGRAMMER-     MAJ ROGER C. WHITON, USAFETAC/DNS, EXT 5412
C*
C*    CREATED ON-     27 FEB 1979          PROJECT-  192301
C*
C*    DESCRIPTION-    THIS SUBROUTINE SUBPROGRAM CALCULATES CLIMATOLOGI-
C*                    CAL WIND FACTORS BY SAWYER'S EQUIVALENT HEADWIND
C*                    TECHNIQUE FOR REASONABLY ARBITRARY GREAT CIRCLE
C*                    ROUTES AT SPECIFIED CONSTANT ALTITUDES FOR ANY OF
C*                    FOUR SEASONS OF THE YEAR IN ANY OF FOUR MODES OR
C*                    WIND OPTIONS (CALM WIND, SIMULATED WIND, MEAN
C*                    WIND, AND 90% WORST WIND).  CLIMB AND DESCENT
C*                    WINDS ARE NOT PROVIDED FOR.  TEMPERATURES AND
C*                    AIRCRAFT PERFORMANCE/FUEL CONSUMPTION ARE NOT PRO-
C*                    VIDED FOR.
C*
C*                    THE SUBPROGRAM NAVIGATES A SIMULATED AIRCRAFT OVER
C*                    A GREAT CIRCLE ROUTE FROM POINT 'A' (FRMLAT, FRMLNG)
C*                    TO POINT 'B' (TOLAT, TOLNG) IN A GLOBAL (LATITUDE/
C*                    LONGITUDE ORIENTED) GRID SYSTEM WHOSE RESOLUTION IS
C*                    15 DEGREES LATITUDE BY 30 DEGREES LONGITUDE.  THE
C*                    SUBROUTINE DETERMINES WHAT GRID SECTOR THE AIRCRAFT
C*                    IS IN AND THE LENGTH OF THE AIRCRAFT'S PATH THROUGH
C*                    THE SECTOR.  THEN THE ROUTINE CONSULTS THE WIND DATA
C*                    BASE FOR THE PARTICULAR GRID SECTOR, FLIGHT LEVEL
C*                    AND SEASON TO OBTAIN SECTOR-MEAN WIND DIRECTION,
C*                    SPEED AND VARIANCE.  CROSS-TRACK AND ALONG-TRACK
C*                    COMPONENTS OF THE WIND ARE COMPUTED.  THESE ARE
C*                    WEIGHTED BY THE LENGTH OF THE AIRCRAFT'S TRACK
C*                    THROUGH THE SECTOR.  AT THE END OF THE SIMULATED
C*                    FLIGHT, A DISTANCE-WEIGHTED, ROUTE-MEAN CROSS-TRACK
C*                    COMPONENT, ALONG-TRACK COMPONENT AND VARIANCE IS
C*                    COMPUTED.  FROM THESE, A ROUTE-MEAN WIND FACTOR IS
C*                    COMPUTED AND ADDED TO THE AIR SPEED TO OBTAIN
C*                    GROUND SPEED.  LATITUDE/LONGITUDE OF POINTS 'A'
C*                    AND 'B,' THE ALTITUDE OF THE AIRCRAFT, THE JULIAN
C*                    DATE AND GMT OF THE FLIGHT, AND THE AIRCRAFT'S
C*                    AIR SPEED ARE GIVEN.  THE GROUND SPEED IS CALCU-
C*                    LATED BASED ON WIND FACTOR.  WIND FACTOR CALCULA-
C*                    TIONS CAN BE MADE IN ANY OF FOUR MODES--CALM WIND,
C*                    SIMULATED WIND, 90% WORST WIND, AND MEAN WIND.
C*                    IN THE CALM WIND CASE, THE ROUTE-MEAN WIND FACTOR
C*                    IS SIMPLY SET TO ZERO, WITH NO NAVIGATION OF THE
C*                    AIRCRAFT OR ACCUMULATION OF DISTANCE-WEIGHTED
C*                    WIND COMPONENTS BEING NECESSARY.  THE SIMULATED
C*                    WIND CASE IS RESERVED FOR FUTURE DEVELOPMENT, IN
C*                    WHICH A VARIABLE WIND MAY BE IMPLEMENTED SUCH
C*                    THAT THE WIND FACTOR NEED NOT BE IDENTICALLY THE
C*                    SAME EVERY TIME THE ROUTE IS FLOWN.  PRESENTLY,
C*                    THE SIMULATED WIND DEFAULTS TO THE MEAN WIND.
C*                    IN THE MEAN WIND CASE, THE MEAN WIND IS DEVELOPED
```

```
C*                  DIRECTLY FROM THE MEAN WIND DATA BASE.  IN THE 90%
C*                  WORST CASE, THIS MEAN WIND FACTOR IS STATISTICALLY
C*                  ADJUSTED TO A VALUE SUCH THAT ONLY 10% OF FLIGHTS
C*                  OVER THAT ROUTE WILL EXPERIENCE A WORSE WIND FACTOR
C*                  (10% RISK).
C*
C*                  THE USER DICTATES WHETHER HE WANTS CALM, SIMULATED,
C*                  MEAN OR 90% WORST WINDS BY SPECIFYING A WIND OPTION
C*                  (IOPTN) FROM 0 TO 3 (SEE TABLE IN PART I OF SUBPRO-
C*                  GRAM).  THE USER SPECIFIES WHICH OF THE FOUR SEA-
C*                  SONS OF THE YEAR HE WANTS WINDS FOR BY STIPULATING
C*                  A JULIAN DATE (JULDAT) FROM 1 TO 366.  THE TABLE FOR
C*                  CONVERSION OF JULIAN DATE TO SEASON OF THE YEAR IS
C*                  PROVIDED IN PART III OF THE SUBPROGRAM.  THE USER
C*                  SPECIFIES THE ALTITUDE OF THE AIRCRAFT BY SETTING
C*                  THE FLAG IALT TO 1 FOR 25,000 FT AND TO 2 FOR
C*                  35,000 FT.  THE USER SPECIFIES HIS AIRSPEED IN
C*                  KNOTS IN THE INPUT VARIABLE ASPEED AND RECEIVES
C*                  CALCULATED GROUND SPEED IN KNOTS IN THE OUTPUT
C*                  VARIABLE GSPEED.
C*
C*                  ARCHITECTURE OF THE SUBPROGRAM IS SUCH THAT PART II
C*                  HANDLES THE CALM WIND CASE, PART III THE MEAN WIND,
C*                  PART IV THE 90% WORST WIND, AND PART V THE SIMULATED
C*                  WIND.  PART I ACCOMPLISHES BRANCHING TO PARTS II-V.
C*                  PART VI HANDLES WRAPUP AND NORMAL TERMINATION.  PART
C*                  VII HANDLES ABNORMAL TERMINATION.
C*
C*    METHOD-       THE METHODS USED BREAK NEATLY INTO TWO CATEGORIES,
C*                  (1) GEODESY AND NAVIGATION, AND (2) METEOROLOGY.
C*
C*                  (1) GEODESY AND NAVIGATION...
C*
C*                  THE AIRCRAFT IS NAVIGATED ALONG A GREAT CIRCLE ROUTE
C*                  BETWEEN POINTS 'A' AND 'B' BY MOVING FROM A POINT
C*                  CALLED THE 'CURRENT-POINT' (XLATC, XLNGC) TO A POINT
C*                  CALLED THE 'NEXT-POINT' (XLATNP, XLNGNP).  MOST OF
C*                  THE COMPUTATION OF THE NEXT-POINT IS DONE IN GLOBAL
C*                  (LATITUDE/LONGITUDE) COORDINATES.
C*
C*                  FOR ANY GIVEN CURRENT-POINT, THE NEXT-POINT IS
C*                  EITHER THE POINT WHERE THE GREAT CIRCLE ROUTE CROS-
C*                  SES A LATITUDE GRID LINE (0, 15, 30, ... DEG) OR
C*                  THE POINT WHERE THE ROUTE CROSSES A LONGITUDE GRID
C*                  LINE (0, 30, 60, ... DEG), WHICHEVER IS CLOSER TO
C*                  THE CURRENT-POINT.
C*
C*                  AT THE CURRENT-POINT, THE AIRCRAFT'S HEADING CAN BE
C*                  CALCULATED VIA THE HEADING FORMULA (FUNCTION HDG).
C*                  KNOWLEDGE OF THE LOCATION OF THE CURRENT-POINT AND
C*                  THE AIRCRAFT'S HEADING AT THAT POINT DICTATES THE
C*                  NEXT LATITUDE CROSSING AND THE NEXT LONGITUDE CROS-
C*                  SING.  CANDIDATE #1 FOR NEXT-POINT IS THE POINT OF
C*                  NEXT LATITUDE CROSSING XLATNX (AND ASSOCIATED LON-
C*                  GITUDE XLNGAS).  CANDIDATE #2 IS THE POINT OF NEXT
C*                  LONGITUDE CROSSING XLNGNX (AND ASSOCIATED LATITUDE
C*                  XLATAS).
C*
C*                  CALCULATION OF LONGITUDE XLNGAS ASSOCIATED WITH A
C*                  GIVEN LATITUDE XLATNX IS DONE ITERATIVELY BY SUB-
C*                  ROUTINE GRTCIR, WHICH SOLVES THE EQUATION OF A
C*                  GREAT CIRCLE.  NEWTON'S ITERATIVE METHOD WITH AN
C*                  EXACT SPHERICAL TRIANGLE FIRST GUESS IS USED.
C*                  VIRTUALLY ALL OF THE COMPLEXITY OF SUBROUTINES
C*                  GRTCIR AND ENRWND IS ATTRIBUTABLE IN ONE WAY OR
C*                  ANOTHER TO THIS PROBLEM OF SOLVING THE EQUATION OF
C*                  A GREAT CIRCLE FOR LONGITUDE, GIVEN THE LATITUDE.
C*
```

```
C*       CALCULATION OF LATITUDE XLATAS ASSOCIATED WITH A
C*       GIVEN LONGITUDE XLNGNX IS DONE DETERMINISTICALLY
C*       BY SUBROUTINE GRTCIR.  NO DIFFICULTY IS ENCOUNTERED
C*       HERE.
C*
C*       CANDIDATES FOR NEXT-POINT ARE THEN AS FOLLOWS...
C*
C*          CANDIDATE #1  ......  (XLATNX, XLNGAS)
C*          CANDIDATE #2  ......  (XLATAS, XLNGNX)
C*
C*       WITH CANDIDATES 1 AND 2 IDENTIFIED, SUBROUTINE
C*       DISTAN IS THEN USED TO CALCULATE GREAT CIRCLE DIS-
C*       TANCES AS FOLLOWS...
C*
C*          S1  ......  GCD FROM CURRENT-POINT TO CAND #1
C*          S2  ......  GCD FROM CURRENT-POINT TO CAND #2
C*
C*       THE CANDIDATE WITH THE SHORTER S-DISTANCE IS SE-
C*       LECTED AS THE NEXT-POINT.
C*
C*       THE CURRENT-POINT CAN THEN BE THOUGHT OF AS THE
C*       POINT WHERE THE AIRCRAFT ENTERS THE GRID SECTOR,
C*       AND THE NEXT-POINT AS THE POINT WHERE THE AIRCRAFT
C*       DEPARTS THE SECTOR.  THE DISTANCE SNP BETWEEN
C*       CURRENT-POINT AND NEXT-POINT IS THE DISTANCE BY
C*       WHICH THE WIND COMPONENTS AND VARIANCE ARE WEIGHTED
C*       IN COMPUTING THE ROUTE-MEAN WIND FACTOR.
C*
C*       2.  METEOROLOGY...
C*
C*       WITH CURRENT-POINT AND NEXT-POINT ESTABLISHED, AN
C*       APPROXIMATE MIDPOINT (XLATM, XLNGM) BETWEEN THEM
C*       IS ESTABLISHED.  THE NUMBER NSEC INDICATING WHICH
C*       OF THE 108 GRID SECTORS THROUGH THE AIRCRAFT IS
C*       PASSING THROUGH IS THEN CALCULATED, BASED ON
C*       (XLATM, XLNGM).  SEE DESCRIPTION OF THE GRID SYS-
C*       TEM BELOW.  SINCE FOUR SEASONS AND TWO FLIGHT
C*       LEVELS, AS WELL AS 108 GRID SECTORS, ARE PROVIDED
C*       FOR, A DATUM-NUMBER NRND IS OBTAINED FROM THE GRID
C*       SECTOR NUMBER NSEC (1-108), THE ALTITUDE INDEX IALT
C*       (1-2) AND THE SEASON INDEX ISEASN (1-4) AS
C*       FOLLOWS...
C*
C*          NRND = (ISEASN-1) * (TOTAL-NUMBER ALTITUDES) *
C*                   (TOTAL-NUMBER-GRID-SECTORS)
C*                   + (IALT-1) * (TOTAL-NUMBER-GRID-SECTORS)
C*                   + NSEC
C*
C*       WITH THE DATUM-NUMBER NRND CALCULATED, THE DATA BASE
C*       (SEE DESCRIPTION BELOW) IS CONSULTED TO OBTAIN SEC-
C*       TOR-MEAN WIND DIRECTION DIRM (RADIANS, IN BETA-ANGLE
C*       FORM, AS DESCRIBED BELOW), SPEED SPDM (KNOTS) AND
C*       VARIANCE VARM (KNOTS**2)...
C*
C*          DIRM  =  DIR(NRND)
C*          SPDM  =  SPD(NRND)
C*          VARM  =  VAR(NRND)
C*
C*       THE AIRCRAFT HEADING AT (XLATM, XLNGM) IS COMPUTED
C*       BY MEANS OF FUNCTION HDG.  THE HEADING HDGM IS COM-
C*       PUTED IN BETA-ANGLE FORMAT (SEE BELOW) AND IN THAT
C*       FORMAT IS REPRESENTED BY THE ANGLE ALPHA.
C*
C*       ALPHA, THE WIND DIRECTION DIRM, AND THE WIND SPEED
C*       SPDM ARE USED TO CALCULATE THE ALONG-GROUND-TRACK
C*       WIND COMPONENT VGM AND THE CROSS-GROUND-TRACK WIND
C*       COMPONENT VCM...
C*
```

```
C*                     VGM = SPDM * COS(ABS(DIRM - ALPHA))
C*                     VCM = SPDM * SIN(    DIRM - ALPHA )
C*
C*                  A TIME-AVERAGED WIND FACTOR FOR THE PRESENT LEG IS
C*                  THEN COMPUTED USING SAWYER'S EQUIVALENT HEADWIND
C*                  FORMULA...
C*
C*                     WBAR = VGM - ((VCM**2 + VARM/2.0) /
C*                            (2.0 * ASPEED))
C*
C*                  BUT NO FURTHER USE IS MADE OF WBAR.
C*
C*                  THE QUANTITIES VGM, VCM AND VARM ARE MULTIPLIED BY
C*                  THE SECTOR- OR LEG-LENGTH SNP, AND THE PRODUCT IS
C*                  ACCUMULATED IN VGACC, VCACC, AND VACC.
C*
C*                  THEN THE NEXT-POINT BECOMES THE CURRENT-POINT, A NEW
C*                  NEXT-POINT IS CALCULATED, AND THE PROCESS IS RE-
C*                  PEATED FOR THE NEXT LEG OF THE SIMULATED FLIGHT.
C*
C*                  AFTER THE LAST LEG OF THE FLIGHT HAS BEEN PROCESSED,
C*                  THE ACCUMULATORS VGACC, VCACC AND VACC ARE DIVIDED
C*                  BY THE TOTAL ROUTE LENGTH.  THE RESULTING DISTANCE-
C*                  WEIGHTED ALONG-TRACK, CROSS-TRACK AND VARIANCE QUAN-
C*                  TITIES ARE THEN USED IN THE SAWYER EQUIVALENT HEAD-
C*                  WIND FORMULA GIVEN ABOVE TO OBTAIN THE DISTANCE-
C*                  WEIGHTED ROUTE-MEAN WIND FACTOR WBARBR.
C*
C*                  IF THE 90% WORST WIND FACTOR HAS BEEN SELECTED,
C*                  WBARBR IS STATISTICALLY ADJUSTED TO THE 90% WORST
C*                  VALUE BY SAWYER'S TECHNIQUE, IMPLEMENTED IN PART IV
C*                  OF SUBROUTINE ENRWND.
C*
C*                  AS PART OF THE WRAPUP ACTIONS IN PART VI OF SUBROU-
C*                  TINE ENRWND, THE ROUTE-MEAN WIND FACTOR WBARBR IS
C*                  ADDED TO THE AIR SPEED ASPEED TO GET A GROUND SPEED
C*                  GSPEED FOR RETURN TO THE PROGRAM CALLING SUBROUTINE
C*                  ENRWND.
C*
C*                  THE METEOROLOGICAL TECHNIQUE USED IS REFERRED TO AS
C*                  SAWYER'S EQUIVALENT HEADWIND TECHNIQUE AND IS DOCU-
C*                  MENTED IN AWS-TR-77-267, GUIDE FOR APPLIED CLIMA-
C*                  TOLOGY.
C*
C*                  BETA-ANGLE FORMAT...
C*
C*                  THE WIND DIRECTION DD CAN BE EXPRESSED IN THE
C*                  FORM OF A BETA-ANGLE B...
C*
C*                           I                          *
C*                           I                      *
C*                           I                  *
C*                           I              *
C*                           I          *
C*                           I      *
C*                           I  *         B
C*                           ------------------------------
C*
C*                  THE BETA ANGLE DESCRIBES THE DIRECTION TOWARD
C*                  WHICH THE WIND IS BLOWING IN TERMS OF RADIANS
C*                  COUNTERCLOCKWISE FROM EASTWARD.  CONVERSIONS
C*                  ARE AS FOLLOWS...
```

```
C*
C*                              DIRECTION        BETA-ANGLE     BETA-ANGLE
C*                            TOWARD WHICH       (DEGREES)      (RADIANS)
C*
C*                                300.0            -30.0         -0.524
C*                                270.0              0.0          0.000
C*                                240.0             30.0          0.524
C*                                180.0             90.0          1.571
C*
C*    GRID
C*    SYSTEM-          THE GRID SYSTEM IS IN GLOBAL (LATITUDE/LONGITUDE)
C*                     COORDINATES, WITH A RESOLUTION OF (15 DEG LAT X
C*                     30 DEG LON).  THE GRID SYSTEM EXTENDS FROM 75 DEG
C*                     N TO 60 DEG S (9 ROWS) AND GLOBALLY IN LONGITUDE
C*                     (12 COLUMNS).  HENCE, THERE ARE 108 GRID SECTORS.
C*                     COLUMN 1 EXTENDS FROM 30 DEG W TO 0 DEG (GREEN-
C*                     WICH), COLUMN 2 FROM 0 DEG TO 30 DEG E, ETC.  ROW
C*                     1 EXTENDS FROM 75 DEG N TO 60 DEG N, ROW 2 FROM
C*                     60 DEG N TO 45 DEG N, ETC.  SUCH A GRID SYSTEM IS
C*                     TOO COARSE FOR CALCULATION OF OPERATIONAL WIND
C*                     FACTORS BUT IS SUITABLE FOR PURPOSES SUCH AS
C*                     SIMULATION.
C*
C*    DATA
C*    BASE-            THE SIOP WINDS WERE USED TO CREATE THE DATA BASE
C*                     EMPLOYED BY SUBROUTINE ENRWND.  SIOP WINDS HAVE
C*                     A (5 DEG X 5 DEG) RESOLUTION.  ALL SIOP WINDS
C*                     WITHIN EACH OF THE PRESENT (15 DEG X 30 DEG) GRID
C*                     SYSTEM WERE EQUALLY WEIGHTED AND AVERAGED TO
C*                     OBTAIN A SINGLE WIND FOR EACH OF THE 108 GRID
C*                     SECTORS, TWO FLIGHT LEVELS (25,000 FT AND
C*                     35,000 FT) AND FOUR SEASONS.  THERE ARE THUS A
C*                     TOTAL OF 864 WINDS IN THE DATA BASE.  EACH HAS A
C*                     WIND DIRECTION DIR(NRND) (RADIANS, BETA-ANGLE
C*                     FORMAT), SPEED SPD(NRND) (KNOTS), AND VARIANCE
C*                     VAR(NRND) (KNOTS**2).  THESE WINDS ARE STORED
C*                     SEQUENTIALLY IN THE FOLLOWING ORDER...
C*
C*                        WINTER SEASON
C*                           FL 250
C*                              108 SECTORS
C*                           FL 350
C*                              108 SECTORS
C*                        SPRING SEASON
C*                           FL 250
C*                              108 SECTORS
C*                           FL 350
C*                              108 SECTORS
C*                              ....
C*                              ETC.
C*                              ....
C*
C*                      THE WINDS ARE IN THE ARRAYS DIR(870), SPD(870)
C*                      AND VAR(870).  THESE ARRAYS ARE MEMBERS OF THE
C*                      LABELLED COMMON BLOCK WEA...
C*
C*                         COMMON /WEA/ DIR(870), SPD(870), VAR(870)
C*
C*                      AN INITIALIZING ROUTINE IS NEEDED TO READ THE
C*                      WIND DATA FROM AN 864-RECORD FILE ON DISK OR
C*                      TAPE TO THE WIND ARRAYS.  THIS INITIALIZATION
C*                      MUST BE EXECUTED BEFORE FIRST CALL TO SUBROUTINE
C*                      ENRWND.
C*
C*    LIMITATIONS-     PRESENTLY, THE WIND FACTOR SIMULATION MODEL CAN-
C*                     NOT PRODUCE A 'SIMULATED' WIND HAVING A DAY-TO-
C*                     DAY VARIABILITY.  HENCE, WITHIN ANY GIVEN SEASON
C*                     AND FOR ANY GIVEN ALTITUDE, A PARTICULAR ROUTE
```

```
C*                          WILL ALWAYS EXPERIENCE THE SAME WIND FACTOR, RE-
C*                          GARDLESS OF THE PASSAGE OF TIME.  REQUESTS FOR
C*                          THE SIMULATED WIND WILL DEFAULT TO THE MEAN
C*                          WIND.
C*
C*                          FURTHERMORE, THE PRESENT MODEL IS INCAPABLE OF
C*                          PRODUCING A 'FORECAST' WIND FACTOR.  REQUESTS
C*                          FOR A FORECAST WIND WILL DEFAULT TO THE MEAN
C*                          WIND.
C*
C*                          TEMPERATURE, AIRCRAFT PERFORMANCE, FUEL CONSUMP-
C*                          TION, AND OTHER FACTORS CONSIDERED BY TYPICAL
C*                          FLIGHT PLANNING MODELS ARE NOT INCLUDED IN THE
C*                          PRESENT MODEL, WHICH DEALS ONLY WITH WIND.
C*
C*                          OPERATIONS NORTH OF 75 DEG N OR SOUTH OF 60 DEG
C*                          S ARE NOT PERMITTED.
C*
C*                          THE COARSENESS OF THE PRESENT 15 DEG LATITUDE BY
C*                          30 DEG LONGITUDE GRID SYSTEM PRECLUDES USE OF
C*                          THE MODEL FOR CALCULATION OF OPERATIONALLY REALIS-
C*                          TIC WIND FACTORS UNLESS CHANGES ARE MADE.  RATHER
C*                          SIMPLE MODIFICATIONS TO THE EXISTING DATA BASE
C*                          AND GRID SYSTEM, ALONG WITH CORRESPONDING CHANGES
C*                          TO THE SOFTWARE, CAN ELIMINATE THIS LIMITATION.
C*
C*                          SPECIFICALLY, THE FOLLOWING RESTRICTIONS MUST BE
C*                          ADHERED TO...
C*
C*                             A.  NO ROUTE SHOULD BE FLOWN DIRECTLY OVER
C*                                 EITHER POLE.
C*
C*                             B.  NEITHER POLE SHOULD SERVE AS A POINT 'A'
C*                                 (TAKEOFF) OR A POINT 'B' (LANDING).
C*
C*                             C.  CIRCUMFERENTIAL OR ROUND ROBIN FLIGHTS IN
C*                                 WHICH POINT 'A' AND 'B' COINCIDE WILL BE
C*                                 ABORTED.  BREAK SUCH FLIGHTS INTO SMALLER
C*                                 SEGMENTS.
C*
C*                             D.  SEMI-CIRCUMFERENTIAL FLIGHTS IN WHICH POINT
C*                                 'B' IS EXACTLY OPPOSITE POINT 'A' THROUGH
C*                                 THE CENTER OF THE EARTH WILL ALSO ABORT.
C*                                 BREAK SUCH FLIGHTS INTO SMALLER SEGMENTS.
C*
C*                             E.  ROUTES OR SEGMENTS OF ROUTES FLOWN DIRECTLY
C*                                 NORTH OR DIRECTLY SOUTH ALONG A LONGITUDE
C*                                 LINE WILL ALSO ABORT.
C*
C*                             F.  AT PRESENT, THE MODEL ALLOWS FLIGHTS ONLY
C*                                 AT ALTITUDES 25,000 FT AND 35,000 FT.
C*
C*                             G.  ALL DEPARTURE AND DESTINATION POINTS MUST
C*                                 LIE WITHIN THE LATITUDINAL RANGE FROM 75
C*                                 DEG N TO 60 DEG S.
C*
C*                          RESTRICTIONS F AND G WERE IMPOSED IN ORDER TO RE-
C*                          DUCE THE CORE STORAGE REQUIREMENTS OF THE MODEL.
C*
C*    REFERENCES-           1.  WHITON, R. C., AND P. L. HEROD, 1980:  WIND FACTOR
C*                              SIMULATION MODEL:  MODEL DESCRIPTION, USAFETAC/TN-80/001.
C*
C*
C*                          2.  HEROD, P. L., AND R. C. WHITON, 1980:  WIND FACTOR
C*                              SIMULATION MODEL:  USER'S MANUAL, USAFETAC/TN-80/002.
C*
```

```
C*
C*   INPUT-         FRMLAT = LATITUDE OF POINT 'A' (TAKEOFF) IN
C*                           DECIMAL DEGREES, NEGATIVE SOUTH
C*                  FRMLNG = LONGITUDE OF POINT 'A' (TAKEOFF) IN
C*                           DECIMAL DEGREES, NEGATIVE EAST
C*                  TOLAT  = LATITUDE OF POINT 'B' (LANDING) IN
C*                           DECIMAL DEGREES, NEGATIVE SOUTH
C*                  TOLNG  = LONGITUDE OF POINT 'B' (LANDING) IN
C*                           DECIMAL DEGREES, NEGATIVE EAST
C*                  JULDAT = JULIAN BASE DATE OF WIND FACTOR RE-
C*                           QUEST, INTEGER, VALUES 1 - 366
C*                  GMT    = GREENWICH BASE TIME OF WIND FACTOR
C*                           REQUEST, VALUE 0.0 - 24.0 HRS
C*                  FCHRS  = FORECAST HOURS AHEAD, VALUE GREATER
C*                           THAN OR EQUAL TO 0.0 HRS
C*                  IALT   = ALTITUDE INDEX, 1 FOR 25,000 FT AND
C*                           2 FOR 35,000 FT
C*                  IOPTN  = WIND OPTION OR MODE, INTEGER, 0 FOR
C*                           CALM WIND, 2 FOR 90% WORST WIND, AND
C*                           3 FOR MEAN WIND.  IN THE FUTURE, 1
C*                           WILL BE IMPLEMENTED AS THE SIMULATED
C*                           WIND BUT PRESENTLY DEFAULTS TO THE
C*                           MEAN WIND
C*                  ASPEED = ROUTE-MEAN AIRCRAFT AIR SPEED IN KNOTS
C*
C*   OUTPUT-        GSPEED = ROUTE-MEAN AIRCRAFT GROUND SPEED IN
C*                           KNOTS
C*
C*   SYSTEM SUB-
C*   PROGRAMS
C*   USED-          SQRT, SIN, COS, FXEM
C*
C*   USER SUB-
C*   PROGRAMS
C*   USED-          GRTCIR, DISTAN, SPHGLO, HDG, BRLAT, BRLNG
C*
C*   ESTIMATED
C*   CPU TIME-      A LONG ROUTE CONSISTING OF 8 LEGS AND REQUIRING
C*                  AN EQUATOR CROSSING TAKES 0.1 - 0.5 SEC CPU TIME
C*                  ON A HONEYWELL 6080 COMPUTER.
C*
C*   STORAGE
C*   REQUIREMENTS-  PROCEDURE PLUS DATA OCCUPY 1130 WORDS OF CORE
C*                  STORAGE.  COMMON BLOCK /WEA/ ADDS 2610 WORDS.
C*                  TOTAL CORE IS 3740 WORDS FOR THIS SUBROUTINE.
C*
C*   PROGRAM
C*   UPDATES-       NONE
C*
C**********************************************************************
C
      COMMON /WEA/ DIR(870), SPD(870), VAR(870)
C     COMMON /DBG/ NDEBUG
C
C     DESCRIPTION OF SPHERICAL GRID.  LATITUDE GRID LINES ARE SPACED
C      EVERY DELTLA DEGREES OF LATITUDE.  LONGITUDE GRID LINES ARE
C      SPACED EVERY DELTLO DEGREES OF LONGITUDE.
C
      DATA DELTLA/15.0/, DELTLO/30.0/
C
C     ANGLE CONVERSION FACTORS.  MULTIPLIER DTOR CONVERTS DEGREES TO
C      RADIANS.  RTOD CONVERTS RADIANS TO DEGREES.
C
      DATA DTOR/0.01745329/, RTOD/57.295780/
C
C     LEGMAX IS MAXIMUM PERMITTED NUMBER OF LEGS IN ROUTE, USED TO
C      DETECT RUNAWAY ROUTES.
C
```

```
      DATA LEGMAX/50/
C
C                  **************************************
C                  *             PART I                 *
C                  *             BRANCH                 *
C                  **************************************
C
C     BRANCH TO APPROPRIATE PART OF SUBROUTINE, BASED ON VALUE OF IOPTN
C      FLAG.
C
C         IOPTN       FUNCTION                  SUBROUTINE PARTS
C
C           0         CALM WIND                 PART II
C           1         SIMULATED WIND            PART V
C           2         90% WORST WIND            PARTS III & IV
C           3         MEAN WIND                 PART III
C
C     NOTE THAT PREPARATION OF THE 90% WORST WIND REQUIRES HAVING PRE-
C      VIOUSLY ESTABLISHED THE MEAN WIND.
C
C     NOTE THAT IN THIS VERSION OF THE SUBROUTINE, THE SIMULATED WIND
C      OPTION IS NOT OPERATIONAL.  CALLS FOR THE SIMULATED WIND WILL DE-
C      FAULT TO THE MEAN WIND.
C
      IF (IOPTN .LT. 0 .OR. IOPTN .GT. 3) GO TO 7020
      IOPTNP = IOPTN + 1
      GO TO (2000, 5000, 3000, 3000), IOPTNP
C
C                  **************************************
C                  *            PART II                 *
C                  *           CALM WIND                *
C                  **************************************
C
C     IN THE CALM WIND CASE, THE WIND FACTOR IS ZERO, AND GROUND SPEED
C      IS EQUAL TO AIR SPEED IN KNOTS. ROUTE-AVERAGED WIND FACTOR WBARBR
C      IS ZERO.
C
 2000 WBARBR = 0.0
C
C     GO TO WRAPUP AND TERMINATION.
C
      GO TO 6000
C
C                  **************************************
C                  *            PART III                *
C                  *           MEAN  WIND               *
C                  **************************************
C
C     INITIALIZE ACCUMULATORS FOR DISTANCE (SACC), VARIANCE (VACC),
C      ALONG-TRACK WIND COMPONENT (VGACC), AND CROSS-TRACK WIND
C      COMPONENT (VCACC).
C
 3000 SACC  = 0.0
      VACC  = 0.0
      VGACC = 0.0
      VCACC = 0.0
C
C     INITIALIZE LAST-LEG FLAG LASTLG.  WHEN THE CURRENT LEG OF THE
C      FLIGHT IS THE LAST LEG, LASTLG = 1.
C
      LASTLG = 0
C
C     TEST POSITION OF POINT 'A' (FRMLAT,FRMLNG) AND POINT 'B' (TOLAT,
C      TOLNG) FOR LEGAL RANGE OF VALUES IN DEGREES.
C
      IF (FRMLAT .LT.  -90.0 .OR. FRMLAT .GT.  90.0) GO TO 7030
      IF (TOLAT  .LT.  -90.0 .OR. TOLAT  .GT.  90.0) GO TO 7030
      IF (FRMLNG .LT. -180.0 .OR. FRMLNG .GT. 180.0) GO TO 7030
```

```
      IF (TOLNG  .LT. -180.0 .OR. TOLNG  .GT. 180.0) GO TO 7030
C
C     INITIALIZE INTERMEDIATE POINT COORDINATES.
C
      XLATC  = 0.0
      XLNGC  = 0.0
      THETAC = 0.0
      PHIC   = 0.0
C
C     INTERPRET THE JULIAN DATE JULDAT IN TERMS OF SEASON OF THE YEAR,
C      INDICATED BY THE SEASON INDEX ISEASN.  GIVE AN ERROR CONDITION
C      FOR ILLEGAL JULDAT VALUES.
C
C         JULIAN DATES          SEASON          SEASON INDEX
C            JULDAT             OF  YEAR            ISEASN
C
C            1 -  59             WINTER               1
C           60 - 151             SPRING               2
C          152 - 243             SUMMER               3
C          244 - 334             FALL                 4
C          335 - 366             WINTER               1
C
      IF (JULDAT .GT. 366) GO TO 7000
      IF (JULDAT .GE. 335) GO TO 3065
      IF (JULDAT .GE. 244) GO TO 3050
      IF (JULDAT .GE. 152) GO TO 3055
      IF (JULDAT .GE.  60) GO TO 3060
      IF (JULDAT .GE.   1) GO TO 3065
      GO TO 7000
 3050 ISEASN = 4
      GO TO 3070
 3055 ISEASN = 3
      GO TO 3070
 3060 ISEASN = 2
      GO TO 3070
 3065 ISEASN = 1
      GO TO 3070
C
C     CHECK ALTITUDE OPTION IALT FOR LEGAL RANGE.
C
C                IALT               AIRCRAFT ALTITUDE
C
C                  1                     25,000 FT
C                  2                     35,000 FT
C
 3070 IF (IALT .LT. 1 .OR. IALT .GT. 2) GO TO 7010
C
C     CALCULATE TOTAL GREAT CIRCLE DISTANCE STOT IN NAUTICAL MILES BE-
C      TWEEN POINTS 'A' AND 'B.'
C
      STOT = 0.0
      CALL DISTAN (FRMLAT, FRMLNG, TOLAT, TOLNG, STOT)
C
C     INITIALIZE GREAT CIRCLE SUBROUTINE.  NOTE THAT GRTCIR IS
C      CALLED TWICE--ONCE FOR ICONV = 1 AND ONCE FOR ICONV = 2.
C
      ICONV =  1
      IGUESS = 0
      XLNGG  = 0.0
      XEAST  = 0.0
      XNORTH = 0.0
      XLATC  = 0.0
      XLNGAP = 0.0
      SAPRIM = 0.0
 3075 CALL GRTCIR (ICONV, FRMLAT, FRMLNG, TOLAT, TOLNG, XLATC, XLNGC,
     &      THETAA, PHIA, THETAB, PHIB, QTX, QTY, QTZ, IGUESS, XLNGG,
     &      XEAST, XNORTH, XLNGAP, SAPRIM, XLATC, THETAC, PHIC)
      IF (ICONV .EQ. 2) GO TO 3080
```

```
      ICONV = 2
      GO TO 3075
C
C     SET THE LOCATION (XLATC,XLNGC) OF THE CURRENT-POINT TO POINT
C      'A' (FRMLAT,FRMLNG).  THIS ROUTINE OPERATES WITH TWO PRINCIPAL
C      POINTERS, NAMELY THE CURRENT-POINT (XLATC,XLNGC) AND THE NEXT-
C      POINT (XLATNP,XLNGNP).  COMPUTATIONS IN THIS SUBROUTINE ARE
C      PRIMARILY IN TERMS OF LATITUDE AND LONGITUDE.  SOME LIMITED
C      USE OF COLATITUDE (THETA) AND AZIMUTH (PHI) IS MADE BY
C      SUBROUTINE GRTCIR, SO THESE QUANTITIES (THETAC, PHIC) ARE
C      MAINTAINED FOR THE CURRENT-POINT (XLATC, XLNGC).
C
 3080 XLATC  = FRMLAT
      XLNGC  = FRMLNG
      ICONVX = 1
      CALL SPHGLO (ICONVX, XLATC, XLNGC, THETAC, PHIC)
C
C     INITIALIZE THE LEG COUNTER LEGNO, WHICH IS UNITY FOR THE FIRST
C      LEG OF THE FLIGHT.
C
      LEGNO = 1
C     IF (NDEBUG .GT. 0) WRITE (6,8000)
C8000 FORMAT (1H0, T8, 'CURR PT', T21, 'NEXT PT', T34, 'DISTANCE', T44,
C      'SEC-', T50, 'ACFT', T55, '-----------WIND----------'/1X,
C      'LG', T7, 'LAT',
C      T14, 'LNG', T20, 'LAT', T27, 'LNG', T33, 'NAUT MILES', T44,
C      'TOR', T51, 'HDG', T58, 'DIR', T64, 'SPD', T70, 'VAR', T74,
C      'FACTOR'//)
C
C     BEGIN NAVIGATION LOOP.
C
C     BASED ON A GREAT CIRCLE BETWEEN THE CURRENT-POINT AND POINT 'B,'
C      CALCULATE THE AIRCRAFT'S HEADING HDGC AT THE CURRENT-POINT.
C      HEADING IS THE DIRECTION TOWARD WHICH IN DEGREES, 0.0 FOR NORTH-
C      WARD, 90.0 FOR EASTWARD, 180.0 FOR SOUTHWARD, AND 270.0 FOR
C      WESTWARD.
C
 3100 HDGC = HDG (XLATC, XLNGC, TOLAT, TOLNG)
C
C     BASED ON THE HEADING HDGC AT THE CURRENT-POINT, DETERMINE THE
C      EASTWARD DIRECTION INDICATOR XEAST (POSITIVE FOR EASTWARD
C      COMPONENT OF AIRCRAFT MOTION) AND NORTHWARD DIRECTION INDI-
C      CATOR XNORTH (POSITIVE FOR NORTHWARD COMPONENT OF AIRCRAFT
C      MOTION).  SPECIAL CASES OF CARDINAL HEADINGS ARE HANDLED BY
C      STATEMENTS 3110 THROUGH 3137, WHILE GENERAL CASES ARE HANDLED
C      BY 3140 THROUGH 3167.
C
 3110 IF (HDGC .EQ.   0.0  .OR.  HDGC .EQ. 360.0) GO TO 3120
      IF (HDGC .EQ.  90.0) GO TO 3125
      IF (HDGC .EQ. 180.0) GO TO 3130
      IF (HDGC .EQ. 270.0) GO TO 3135
      GO TO 3140
 3120 XEAST  = 0.0
      XNORTH = 1.0
      GO TO 3170
 3125 XEAST  = 1.0
      XNORTH = 0.0
      GO TO 3170
 3130 XEAST  = 0.0
      XNORTH =-1.0
      GO TO 3170
 3135 XEAST  =-1.0
      XNORTH = 0.0
 3137 GO TO 3170
 3140 IF (HDGC .LT.  90.0) GO TO 3150
      IF (HDGC .LT. 180.0) GO TO 3155
      IF (HDGC .LT. 270.0) GO TO 3160
      GO TO 3165
```

```
      3150 XEAST  =  1.0
           XNORTH =  1.0
           GO TO 3170
      3155 XEAST  =  1.0
           XNORTH = -1.0
           GO TO 3170
      3160 XEAST  = -1.0
           XNORTH = -1.0
           GO TO 3170
      3165 XEAST  = -1.0
      3167 XNORTH =  1.0
      3170 CONTINUE
C          IF (NDEBUG .GT. 1) WRITE (6,8010) XEAST, XNORTH
C8010 FORMAT (1H , T4, 'XEAST,XNORTH = ', T36, 2F10.4)
C
C          BASED ON EASTWARD DIRECTION INDICATOR, CALCULATE THE POINT OF NEXT
C           LONGITUDE CROSSING (XLNGNX) AND ASSOCIATED LATITUDE (XLATAS) DE-
C           TERMINED FROM THE EQUATION OF A GREAT CIRCLE.  THIS IS CANDIDATE
C           POINT #1.  STATEMENTS 3200-3299.  DIFFERENT PROCEDURE IS USED,
C           DEPENDING ON WHETHER THIS IS THE FIRST LEG OF THE FLIGHT OR
C           WHETHER (ON SUBSEQUENT LEGS) THE CURRENT-POINT WAS CANDIDATE #1
C           (NCAND = 1) OR CANDIDATE #2 (NCAND = 2).
C
      3200 IF (LEGNO .GT. 1) GO TO 3240
C
C          PROCEDURE FOR FIRST LEG OF FLIGHT.  DETERMINE BRACKETING LONGI-
C           TUDE VALUES.  MOVE EAST (DECREASING LONGITUDE) IF XEAST .GT.
C           0.0.  SET XLNGNX.
C
      3210 CALL BRLNG (XLNGC, XLNGLO, XLNGHI)
C          IF (NDEBUG .GT. 1) WRITE (6,8020) XLNGLO, XLNGHI
C8020 FORMAT (1H , T4, 'XLNGLO,XLNGHI = ', T36, 2F10.4)
           IF (XEAST) 3220,3220,3230
      3220 XLNGNX = XLNGHI
           GO TO 3290
      3230 XLNGNX = XLNGLO
           GO TO 3290
C
C          PROCEDURE FOR SUCCEEDING LEGS DEPENDS ON WHETHER CANDIDATE #1 OR
C           CANDIDATE #2 WAS CHOSEN LAST TIME.
C
      3240 GO TO (3250,3270), NCAND
C
C          IF CURRENT-POINT WAS A CANDIDATE #1 (EXACT LONGITUDE), THEN NEXT
C           LONGITUDE CROSSING IS THAT LONGITUDE PLUS OR MINUS DELTA-LONGI-
C           TUDE (PLUS FOR WESTWARD, MINUS FOR EASTWARD).
C
      3250 XLNGNX = -(XEAST * DELTLO) + XLNGC
C
C          ADJUST LONGITUDE FOR DATELINE FOLD.
C
           IF (XLNGNX .GT.  180.0) GO TO 3260
           IF (XLNGNX .LT. -180.0) GO TO 3265
           GO TO 3290
      3260 XLNGNX = XLNGNX - 360.0
           GO TO 3290
      3265 XLNGNX = XLNGNX + 360.0
           GO TO 3290
C
C          IF CURRENT-POINT WAS A CANDIDATE #2 (INEXACT LONGITUDE), THEN
C           NEXT LONGITUDE CROSSING MUST BE FOUND BY THE BRACKETING METHOD
C           USED FOR LEGNO = 1.
C
      3270 GO TO 3210
C
C          SET ASSOCIATED LATITUDE (XLATAS).
C
      3290 ICONV = 3
```

```
            CALL GRTCIR (ICONV, FRMLAT, FRMLNG, TOLAT, TOLNG, XLATAS, XLNGNX,
     &       THETAA, PHIA, THETAB, PHIB, QTX, QTY, QTZ, IGUESS, XLNGG,
     &       XEAST, XNORTH, XLNGAP, SAPRIM, XLATC, THETAC, PHIX)
C           IF (NDEBUG .GT. 1) WRITE (6,8030) XLATAS, XLNGNX
C8030 FORMAT (1H , T4, 'XLATAS,XLNGNX = ', T36, 2F10.4)
C
C           BASED ON NORTHWARD DIRECTION INDICATOR, CALCULATE THE POINT OF
C            NEXT LATITUDE CROSSING (XLATNX) AND ASSOCIATED LONGITUDE
C            (XLNGAS) DETERMINED FROM THE EQUATION OF A GREAT CIRCLE.  THIS
C            IS CANDIDATE POINT #2.  STATEMENTS 3300-3399.  DIFFERENT PRO-
C            CEDURE IS USED, DEPENDING WHETHER THIS IS FIRST LEG OF FLIGHT
C            OR WHETHER (ON SUBSEQUENT LEGS) THE CURRENT-POINT WAS A
C            CANDIDATE #1 (NCAND = 1) OR A CANDIDATE #2 (NCAND = 2).
C
C           FIRST SET COUNTER OF 'GRTCIR' CONVERGENCE FAILURES TO ZERO.
C            THIS WILL LATER BE USED TO REVERSE THE LATITUDE SEARCH.
C
 3300 NFAIL = 0
C
C           DISCRIMINATE FIRST FROM SUBSEQUENT LEGS.
C
 3310 IF (LEGNO .GT. 1) GO TO 3340
C
C           PROCEDURE FOR FIRST LEG.  DETERMINE BRACKETING LATITUDE VALUES.
C            MOVE NORTH (INCREASING LATITUDE) IF XNORTH .GT. 0.0.  SET
C            XLATNX.
C
 3315 CALL BRLAT (XLATC, XLATLO, XLATHI)
C           IF (NDEBUG .GT. 1) WRITE (6,8040) XLATLO, XLATHI
C8040 FORMAT (1H , T4, 'XLATLO,XLATHI = ', T36, 2F10.4)
            IF (XNORTH) 3320,3320,3330
 3320 XLATNX = XLATLO
            GO TO 3390
 3330 XLATNX = XLATHI
            GO TO 3390
C
C           PROCEDURE FOR SUCCEEDING LEGS DEPENDS ON WHETHER CANDIDATE #1 OR
C            CANDIDATE #2 WAS CHOSEN LAST TIME.
C
 3340 GO TO (3350,3370), NCAND
C
C           IF CURRENT-POINT WAS A CANDIDATE #1 (INEXACT LATITUDE), THEN NEXT
C            LATITUDE CROSSING MUST BE FOUND BY THE BRACKETING METHOD USED FOR
C            LEGNO = 1.
C
 3350 GO TO 3315
C
C           IF CURRENT-POINT WAS A CANDIDATE #2 (EXACT LATITUDE), THEN NEXT
C            LATITUDE CROSSING IS THAT LATITUDE PLUS OR MINUS DELTA-LATITUDE
C            (PLUS FOR NORTHWARD, MINUS FOR EASTWARD).
C
 3370 XLATNX = (XNORTH * DELTLA) + XLATC
            IF ( ABS(XLATNX) .LE. 0.001 )  XLATNX = 0.0
C
C           ADJUST LATITUDE FOR POLAR SINGULARITY.
C
            IF (XLATNX .GT.  90.0) GO TO 3375
            IF (XLATNX .LT. -90.0) GO TO 3380
            GO TO 3390
 3375 XLATNX =  180.0 - XLATNX
            GO TO 3390
 3380 XLATNX = -180.0 - XLATNX
            GO TO 3390
C
C           ATTEMPT TO SET ASSOCIATED LONGITUDE (XLNGAS).
C
 3390 ICONV = 5
C           IF (NDEBUG .GT. 1) WRITE (6,8050) XLATNX
```

```
C8050 FORMAT (1H , T4, 'XLATNX         = ', T36, F10.4)
      CALL GRTCIR (ICONV, FRMLAT, FRMLNG, TOLAT, TOLNG, XLATNX,
&      XLNGAS, THETAA, PHIA, THETAB, PHIB, QTX, QTY, QTZ,
&      IGUESS, XLNGG, XEAST, XNORTH, XLNGAP, SAPRIM, XLATC, THETAX,
&      PHIC)
C
C     RESCUE PHIC FOR LATER USE.
C
      ICONVX = 1
      CALL SPHGLO (ICONVX, XLATC, XLNGC, THETAC, PHIC)
C     IF (NDEBUG .GT. 1) WRITE (6,8060) ICONV
C8060 FORMAT (1H , T4, 'ICONV          = ', T36, I10)
C     IF (NDEBUG .GT. 1) WRITE (6,8070) XLNGG
C8070 FORMAT (1H , T4, 'FIRST GUESS LONGITUDE = ', T36, F10.4)
C     IF (NDEBUG .GT. 1) WRITE (6,8080) XLATNX, XLNGAS
C8080 FORMAT (1H , T4, 'XLATNX,XLNGAS = ', T36, 2F10.4)
C
C     IF 'GRTCIR' SUBROUTINE FAILS TO CONVERGE IN LONGITUDE, INCREMENT
C      FAILURE COUNTER BY ONE, REVERSE LATITUDE SEARCH DIRECTION, AND
C      TRY AGAIN TO DETERMINE THE NEXT LATITUDE CROSSING (XLATNX) AND
C      ASSOCIATED LONGITUDE.  OTHERWISE, CONTINUE.  TWO CONSECUTIVE
C      FAILURES PRODUCE AN ERROR ABORT.
C
      IF (ICONV .NE. 11) GO TO 3400
      NFAIL = NFAIL + 1
      IF (NFAIL .GT. 1) GO TO 7040
      XNORTH = -XNORTH
C     IF (NDEBUG .GT. 1) WRITE (6,8090) XNORTH
C8090 FORMAT (1H , T4, 'REVERSED XNORTH = ', T36, F10.4)
      GO TO 3310
C
C     CALCULATE GREAT CIRCLE DISTANCE (GCD) S1 FROM CURRENT-POINT
C      (XLATC,XLNGC) TO CANDIDATE #1 (XLATAS,XLNGNX).  CALCULATE
C      GCD S2 FROM CURRENT-POINT TO CANDIDATE #2 (XLATNX,XLNGAS).
C
 3400 CALL DISTAN (XLATC, XLNGC, XLATAS, XLNGNX, S1)
      CALL DISTAN (XLATC, XLNGC, XLATNX, XLNGAS, S2)
C
C     WHICHEVER OF THE TWO POINTS, CANDIDATES 1 OR 2, IS CLOSEST TO THE
C      CURRENT-POINT IS SELECTED AS THE NEXT-POINT (XLATNP,XLNGNP)
C      ALONG THE PATH.  DECISION IS MADE ON THE BASIS OF GCD S1 FOR CAN-
C      DIDATE POINT #1 AND S2 FOR CANDIDATE POINT #2.
C
      IF (S2 - S1) 3420,3420,3430
C
C     CASE OF S1 .GE. S2 ... CHOOSE CANDIDATE #2.
C
 3420 NCAND  = 2
      XLATN  = XLATNP
      XLNGN  = XLNGNP
      XLATNP = XLATNX
      XLNGNP = XLNGAS
      SNP    = S2
      GO TO 3450
C
C     CASE OF S2 .GE. S1 ... CHOOSE CANDIDATE #1.
C
 3430 NCAND  = 1
      XLATN  = XLATNP
      XLNGN  = XLNGNP
      XLATNP = XLATAS
      XLNGNP = XLNGNX
      SNP    = S1
C
C     NOTE...  (XLATN,XLNGN) IS CROSSING POINT BEFORE NEXT-POINT.
C
C     CALCULATE GREAT CIRCLE DISTANCE (GCD) SNEXT IN NAUTICAL MILES
C      FROM POINT 'A' (FRMLAT,FRMLNG) TO THE NEXT-POINT (XLATNP,
```

```
C       XLNGNP) JUST SELECTED.  IF THAT DISTANCE (SNEXT) IS GREATER
C       THAN THE TOTAL GCD (STOT) BETWEEN 'A' AND 'B,' THEN THE NEXT-
C       POINT IS BEYOND THE TERMINATION-POINT 'B.'  IN SUCH A CASE,
C       THE COORDINATES OF 'B' MUST BE USED AS THE LAST-POINT
C       (XLATNP,XLNGNP), DISTANCE SNP MUST BE ADJUSTED, AND LAST-LEG
C       FLAG MUST BE TURNED ON.
C
 3450 SNEXT = 0.0
C     IF (NDEBUG .GT. 1) WRITE (6,8100) NCAND, XLATNP, XLNGNP
C8100 FORMAT (1H , T4, 'NCAND,XLATNP,XLNGNP = ', T36, I10, 2F10.4)
      CALL DISTAN (FRMLAT, FRMLNG, XLATNP, XLNGNP, SNEXT)
      IF (STOT - SNEXT) 3460,3500,3500
C
C     PROCEDURE FOR SNEXT .GT. STOT.
C
 3460 LASTLG  = 1
      XLATNP  = TOLAT
      XLNGNP  = TOLNG
      CALL DISTAN (XLATN, XLNGN, TOLAT, TOLNG, SNP)
C
C     FIND APPROXIMATE MIDPOINT BETWEEN CURRENT-POINT (XLATC,XLNGC)
C      AND NEXT-POINT (XLATNP,XLNGNP).  THIS MIDPOINT IS LATER USED
C      TO DETERMINE GRID SECTOR NUMBER NSEC AND AS POSITION FOR
C      COMPUTATION OF AIRCRAFT HEADING HDGM.
C
 3500 XLATM  =  0.5 * (XLATC + XLATNP)
      XLNGM  =  0.5 * (XLNGC + XLNGNP)
C
C     FIND ROW NUMBER NROW (LATITUDE COUNTER) AND COLUMN NUMBER NCOL
C      (LONGITUDE COUNTER) OF MIDPOINT.  THEN USE NROW AND NCOL TO
C      COMPUTE SECTOR NUMBER NSEC IN WHICH THE MIDPOINT (XLATM,
C      XLNGM) LIES.  IF POINT LIES OUTSIDE PLUS OR MINUS 75.0 DEGREES
C      LATITUDE LIMITS OF THE GRID SYSTEM, BOGUS THE POINT INTO THE
C      SYSTEM.
C
      XLATDM = XLATM
      IF (XLATDM .GT.  75.0) XLATDM =  75.0
      IF (XLATDM .LT. -60.0) XLATDM = -60.0
      ANGLE  = 75.0 - XLATDM
      NROW   = (ANGLE / DELTLA) + 1
      ANGLE  = -XLNGM + 30.0
      IF (XLNGM .GT. 0.0) ANGLE = 390.0 - XLNGM
      NCOL   = (ANGLE / DELTLO) + 1
      IF (NCOL .GT. 12) NCOL = NCOL - 12
      NSEC   = ((NROW - 1) * 12) + 1 + (NCOL - 1)
C
C     CALCULATE DATUM NUMBER NRND BASED ON SECTOR NUMBER NSEC
C      (1-108), AIRCRAFT ALTITUDE IALT (1-2) AND SEASON INDEX
C      ISEASN (1-4).
C
      NRND = (ISEASN - 1) * 2 * 108  +  (IALT - 1) * 108  +  NSEC
C
C     CALCULATE AIRCRAFT HEADING HDGM (DIRECTION TOWARD WHICH IN
C      DEGREES) AT THE MIDPOINT 'M.'
C
      HDGM = HDG (XLATM, XLNGM, TOLAT, TOLNG)
C
C     CONVERT HEADING IN DEGREES TO ANGLE ALPHA IN RADIANS.  ALPHA
C      IS THE DIRECTION TOWARD WHICH THE AIRCRAFT'S GROUND TRACK IS
C      INCLINED, IN RADIANS COUNTERCLOCKWISE FROM EASTWARD.
C
      ALPHA =  90.0 - HDGM
      IF (ABS(HDGM) - 180.0) 3560,3560,3550
 3550 ALPHA = 360.0 + ALPHA
 3560 ALPHA = DTOR  * ALPHA
C
C     RETRIEVE DIRECTION DIRM AND SPEED SPDM TOWARD WHICH THE WIND IS
C      BLOWING AT THE MIDPOINT POSITION (XLATM,XLNGM).  DIRECTION IS
```

```
C        IN RADIANS COUNTERCLOCKWISE FROM EASTWARD (E.G., WIND FROM 225
C        DEGREES HAS DIRM = +PI/4 = 0.78540).  SPEED IS IN KNOTS.  ALSO
C        RETRIEVE VECTOR STANDARD VARIANCE VARM (SQUARE OF VECTOR STAN-
C        DARD DEVIATION) IN UNITS OF KNOTS**2.
C
      DIRM = DIR(NRND)
      SPDM = SPD(NRND)
      VARM = VAR(NRND)
C
C     COMPUTE ALONG-GROUND-TRACK WIND COMPONENT VGM AND CROSS-GROUND-
C      TRACK WIND COMPONENT VCM IN KNOTS.
C
      GAMMA  =  DIRM - ALPHA
      GAMMAA =  ABS(GAMMA)
      VGM    =  SPDM * COS(GAMMAA)
      VCM    =  SPDM * SIN(GAMMA )
C
C     COMPUTE TIME-AVERAGED WIND FACTOR WBAR IN KNOTS FOR THE
C      PRESENT LEG.
C
      WBAR = VGM - ((VCM*VCM + (VARM/2.0)) / (2.0*ASPEED))
C     IF (NDEBUG .GT. 0) WRITE (6,8110) LEGNO, XLATC, XLNGC, XLATNP,
C      XLNGNP, SNP, NSEC, HDGM, DIRM, SPDM, VARM, WBAR
C8110 FORMAT (1H ,I2,2(F6.1,F7.1),T33,F8.2,T42,I4,1X,2F7.3,F6.1,
C      F6.0, F7.1)
C
C     ACCUMULATE TOTAL DISTANCE TO THE ACCUMULATOR SACC, DISTANCE-
C      WEIGHTED TOTAL VARIANCE TO VACC, DISTANCE-WEIGHTED TOTAL
C      ALONG-TRACK COMPONENT TO VGACC, AND DISTANCE-WEIGHTED
C      TOTAL CROSS-TRACK COMPONENT TO VCACC.
C
      SACC  = SACC  + SNP
      VACC  = VACC  + (SNP * VARM)
      VGACC = VGACC + (SNP * VGM )
      VCACC = VCACC + (SNP * VCM )
C
C     ADVANCE TO NEXT-POINT.  OLD NEXT-POINT (XLATNP,XLNGNP) BECOMES
C      CURRENT-POINT (XLATC,XLNGC).  UPDATE (THETAC, PHIC) COR-
C      RESPONDING TO (XLATC, XLNGC).  INCREMENT LEG NUMBER COUNTER.
C      GIVE ERROR CONDITION FOR RUNAWAY ROUTE (EXCESSIVE NUMBER OF
C      LEGS).
C
      XLATC  = XLATNP
      XLNGC  = XLNGNP
      ICONVX = 1
      CALL SPHGLO (ICONVX, XLATC, XLNGC, THETAC, PHIC)
      LEGNO  = LEGNO + 1
      IF (LEGNO .GT. LEGMAX) GO TO 7050
C
C     IF CURRENT-POINT (XLATC,XLNGC) IS POINT 'B' (TOLAT,TOLNG), THIS
C      IS THE LAST LEG OF THE FLIGHT, AND SUBROUTINE BRANCHES TO WRAP-
C      UP.  OTHERWISE, ROUTINE GOES BACK FOR THE NEXT LEG.
C
      XLADIF = ABS(XLATC - TOLAT)
      XLNDIF = ABS(XLNGC - TOLNG)
      IF (XLADIF .LT. 0.0001 .AND. XLNDIF .LT. 0.0001) LASTLG = 1
      IF (LASTLG .EQ. 0) GO TO 3100
C
C     COMPUTE FINAL ROUTE-AVERAGED WIND FACTOR WBARBR.  DIVIDE DISTANCE-
C      WEIGHTED ACCUMULATORS BY TOTAL DISTANCE TO GET ROUTE-MEAN VARI-
C      ANCE VARBP, ROUTE-MEAN ALONG-TRACK COMPONENT VGBR, AND ROUTE-
C      MEAN CROSS-TRACK COMPONENT VCBR.  THEN CALCULATE THE ROUTE-MEAN
C      WIND FACTOR WBARBR.
C
      VARBR  = VACC  / SACC
      VGBR   = VGACC / SACC
      VCBR   = VCACC / SACC
      WBARBR = VGBR - ((VCBR*VCBR + (VARBR / 2.0)) / (2.0 * ASPEED))
```

```
C     IF (NDEBUG .GT. 0) WRITE (6,8120) STOT, VGBR, VCBR, VARBR, WBARBR
C8120 FORMAT (1H0, 'GREAT CIRCLE DISTANCE', T30, F15.5//1X,
C      'ROUTE-AVERAGE', T16, 'ALONG-TRACK', T30, F15.5/
C      T16, 'CROSS-TRACK', T30, F15.5/T16, 'VARIANCE', T30, F15.5/
C      T16, 'WIND FACTOR', T30, F15.5)
C
C     IF IOPTN = 2, BRANCH TO PART IV OF THE SUBROUTINE TO ADJUST THE
C      WIND FACTOR TO THE 90% WORST CASE.  OTHERWISE, BRANCH TO PART
C      VI FOR WRAPUP AND NORMAL TERMINATION.
C
      GO TO (6000, 6000, 4000, 6000), IOPTNP
C
C                    **************************************
C                    *               PART IV              *
C                    *           90% WORST WIND           *
C                    **************************************
C
C     90% WORST WIND FACTOR CALCULATION EMPLOYS THE SAWYER METHOD
C      DESCRIBED IN AIR WEATHER SERVICE TECHNICAL REPORT 77-267,
C      GUIDE TO APPLIED CLIMATOLOGY, PAGE 6-8.  THE NEW VARIABLES
C      USED IN THIS CALCULATION ARE AS FOLLOWS...
C
C        XKFACT = FACTOR TO CONVERT MEAN STANDARD VECTOR DEVIATION
C                 OF WINDS OVER A ROUTE (STVDVW) TO STANDARD DEVIA-
C                 TION OF THE MEAN WIND FACTOR (STDVWF)
C
C        STVDVW = MEAN STANDARD VECTOR DEVIATION OF WINDS OVER A
C                 ROUTE
C
C        STDVWF = STANDARD DEVIATION OF THE MEAN WIND FACTOR
C
C        WBARBR = 90% WORST WIND FACTOR (I.E., 10% RISK)
C
C     CALCULATE K-FACTOR OF SAWYER.  THE RELATION USED BELOW IS A
C      GEOMETRIC CURVE FIT TO SAWYER'S DATA REPORTED IN AWSTR 77-
C      267.  THE GEOMETRIC CURVE FIT WAS CONSTRUCTED USING HEWLETT-
C      PACKARD 65 STAT PAC #2, PAGE 22.  THE FUNCTION GIVES
C      K (XKFACT) AS A FUNCTION OF ROUTE LENGTH (STOT).
C
 4000 XKFACT = 0.70585066 * (0.99983056**STOT)
C
C     CALCULATE MEAN STANDARD VECTOR DEVIATION OF WIND OVER ROUTE.
C
      STVDVW = SQRT(VARBR)
C
C     CALCULATE STANDARD DEVIATION OF THE MEAN WIND FACTOR.
C
      STDVWF = STVDVW * XKFACT
C
C     CALCULATE 90% WORST WIND FACTOR.
C
      WBARBR = WBARBR - (1.28 * STDVWF)
C
C     WRAPUP AND NORMAL TERMINATION.
C
      GO TO 6000
C
C                    **************************************
C                    *               PART V               *
C                    *           SIMULATED WIND           *
C                    **************************************
C
C     SIMULATED WIND PRESENTLY DEFAULTS TO THE MEAN WIND CASE.
C
 5000 GO TO 3000
C
```

```
C                    **************************************
C                    *                PART VI             *
C                    *    WRAPUP & NORMAL TERMINATION     *
C                    **************************************
C
C     WIND FACTOR IS GROUND SPEED MINUS AIR SPEED.  USE THIS RELATION
C      TO COMPUTE GROUND SPEED (GSPEED) IN KNOTS FROM GIVEN AIR SPEED
C      (ASPEED) IN KNOTS AND ROUTE-AVERAGED WIND FACTOR (WBARBR) IN
C      KNOTS.
C
 6000 GSPEED = ASPEED + WBARBR
C
C     NORMAL TERMINATION.
C
      RETURN
C
C                    **************************************
C                    *                PART VII            *
C                    *         ABNORMAL TERMINATION       *
C                    **************************************
C
 7000 GSPEED = ASPEED
      WRITE (6,8130) JULDAT, GMT, IALT, FRMLAT, FRMLNG, TOLAT,
     &       TOLNG, IOPTN
 8130 FORMAT (1H0, '****WIND FACTOR ERROR DIAGNOSTICS****'/1X,
     &       'JULIAN DATE = ', T15, I10, T35, 'GMT = ', T41, F10.2, T53,
     &       'ALT OPTION = ', T66, I10/1X,
     &       'FROM-LAT    = ', F10.2, T30, 'FROM-LNG = ', T41, F10.2/1X,
     &       'TO-LAT      = ', F10.2, T32, 'TO-LNG = ', F10.2, T52,
     &       'WIND OPTION = ', I10//)
      CALL FXEM (61, 'SUBROUTINE ENRWND...ILLEGAL JULIAN DATE', 10)
      RETURN
 7010 GSPEED = ASPEED
      WRITE (6,8130) JULDAT, GMT, IALT, FRMLAT, FRMLNG, TOLAT,
     &       TOLNG, IOPTN
      CALL FXEM (61, 'SUBROUTINE ENRWND...ILLEGAL ALTITUDE', 9)
      RETURN
 7020 GSPEED = ASPEED
      WRITE (6,8130) JULDAT, GMT, IALT, FRMLAT, FRMLNG, TOLAT,
     &       TOLNG, IOPTN
      CALL FXEM (61, 'SUBROUTINE ENRWND...ILLEGAL WIND OPTION', 10)
      RETURN
 7030 GSPEED = ASPEED
      WRITE (6,8130) JULDAT, GMT, IALT, FRMLAT, FRMLNG, TOLAT,
     &       TOLNG, IOPTN
      CALL FXEM (61, 'SUBROUTINE ENRWND...ILLEGAL LAT/LON', 9)
      RETURN
 7040 GSPEED = ASPEED
      WRITE (6,8130) JULDAT, GMT, IALT, FRMLAT, FRMLNG, TOLAT,
     &       TOLNG, IOPTN
      CALL FXEM (61, 'SUBROUTINE ENRWND...GRTCIR FAILED TWICE', 10)
 7050 GSPEED = ASPEED
      WRITE (6,8130) JULDAT, GMT, IALT, FRMLAT, FRMLNG, TOLAT,
     &       TOLNG, IOPTN
      CALL FXEM (61, 'SUBROUTINE ENRWND...RUNAWAY ROUTE', 9)
      RETURN
      END
```

```
CGRTCIR EQ GRT CIRCLE/R. C. WHITON/05 FEB 1979
C
      SUBROUTINE GRTCIR (ICONV, FRMLAT, FRMLNG, TOLAT, TOLNG, XLAT,
     &  XLNG, THETAA, PHIA, THETAB, PHIB, QTX, QTY, QTZ, IGUESS, XLNGG,
     &  XEAST, XNORTH, XLNGAP, SAPRIM, XLATC, THETA, PHI)
C
C*********************************************************************
C*
C*   PROGRAM ID-    GRTCIR
C*   MET ANALYST-   MAJ ROGER C. WHITON, USAFETAC/DNS, EXT 5412
C*   SYS ANALYST-   MAJ ROGER C. WHITON, USAFETAC/DNS, EXT 5412
C*   PROGRAMMER-    MAJ ROGER C. WHITON, USAFETAC/DNS, EXT 5412
C*
C*   CREATED ON-    05 FEB 1979           PROJECT-   192301
C*
C*   DESCRIPTION-   THIS SUBROUTINE SUBPROGRAM SOLVES THE EQUATION OF
C*                  THE GREAT CIRCLE DEFINED BY POINTS 'A' ((FRMLAT,
C*                  FRMLNG) OR (THETAA,PHIA)) AND 'B' ((TOLAT,TOLNG)
C*                  OR (THETAB,PHIB)) AT AN INTERMEDIATE POINT
C*                  ((XLAT,XLNG) OR (THETA,PHI)).  DEPENDING ON THE
C*                  VALUE OF THE FLAG ICONV, THE PROGRAM PERFORMS ONE
C*                  OF THE FOLLOWING FUNCTIONS...
C*
C*                  ICONV  FUNCTION
C*
C*                    1    GIVEN INPUT LATITUDE AND LONGITUDE OF POINTS
C*                         'A' (FRMLAT,FRMLNG) AND 'B' (TOLAT,TOLNG),
C*                         CALCULATE SPHERICAL COORDINATES (THETAA,
C*                         PHIA) AND (THETAB,PHIB) OF THE POINTS BY
C*                         INVOKING SUBROUTINE SPHGLO.  THEN COMPUTE
C*                         THE CROSS PRODUCT VECTOR  QT = QA X QB OF
C*                         'A' AND 'B.'
C*
C*                    2    INITIALIZE THE EXACT SPHERICAL TRIANGLE
C*                         METHOD OF OBTAINING FIRST GUESS LONGITUDE.
C*                         FIRST, DETERMINE THE LONGITUDE XLNGAP OF
C*                         THE POINT A-PRIME AT WHICH THE GREAT CIR-
C*                         CLE ROUTE CROSSES THE EQUATOR.  THEN ES-
C*                         TABLISH THE INTERIOR ANGLE A-PRIME OF THE
C*                         SPHERICAL TRIANGLE (A-PRIME,C-PRIME,B).
C*                         THE SINE OF THAT ANGLE IS SAPRIM.
C*
C*                    3    GIVEN INPUT INTERMEDIATE LONGITUDE XLNG,
C*                         CALCULATE INTERMEDIATE PHI, INTERMEDIATE
C*                         THETA AND INTERMEDIATE LATITUDE XLAT.
C*
C*                    4    GIVEN INPUT INTERMEDIATE PHI, CALCULATE IN-
C*                         TERMEDIATE THETA AND INTERMEDIATE LATITUDE
C*                         XLAT.
C*
C*                    5    GIVEN INPUT INTERMEDIATE LATITUDE XLAT,
C*                         CALCULATE INTERMEDIATE THETA, INTERMEDIATE
C*                         PHI AND INTERMEDIATE LONGITUDE XLNG.
C*
C*                    6    GIVEN INPUT INTERMEDIATE THETA, CALCULATE
C*                         INTERMEDIATE PHI AND INTERMEDIATE LONGI-
C*                         TUDE XLNG.
C*
C*                  THIS SUBROUTINE MUST BE INVOKED ONCE FOR FUNCTION
C*                  ICONV = 1 AND ONCE FOR ICONV = 2 BEFORE BEING
C*                  USED FOR FUNCTION ICONV = 3 THROUGH 6.  THUS
C*                  THERE MUST BE  'INITIAL' CALLS TO THIS ROUTINE
C*                  USING ICONV = 1 AND 2 BEFORE CALLS USING
C*                  ICONV = 3 THROUGH 6.
C*
C*   METHOD-        THE METHOD USED EMERGES FROM THE VECTOR CALCULUS
C*                  IN A RECTANGULAR (X,Y,Z) COORDINATE SYSTEM.  QA IS
C*                  THE POSITION VECTOR OF POINT 'A,' WHILE QB IS THAT
```

```
C*                     OF POINT 'B.'  THE POSITION VECTOR OF THE INTER-
C*                     MEDIATE POINT IS Q.  THE PLANE OF THE GREAT CIRCLE
C*                     IS DEFINED BY THE CROSS PRODUCT  QT = QA X QB,
C*                     WHICH IS DIRECTED NORMAL TO THE PLANE.  THE EQUA-
C*                     TION OF THE GREAT CIRCLE IS DEFINED BY THE RELATION,
C*
C*                                Q . (QA X QB)  =  0
C*
C*                     WHERE THE PERIOD INDICATES A DOT PRODUCT.  THIS IS
C*                     EQUIVALENT TO SAYING THAT THE COMPONENT OF Q IN THE
C*                     QT-DIRECTION IS ZERO.  EXPANSION OF THIS PRODUCT IN
C*                     RECTANGULAR COORDINATES LEADS TO THE RELATION,
C*
C*                       QTX*SIN(THETA)*COS(PHI) + QTY*SIN(THETA)*SIN(PHI)
C*                          + QTZ*COS(THETA)  =  0
C*
C*                     WHERE QTX, QTY AND QTZ ARE THE X, Y AND Z-COMPONENTS
C*                     OF THE CROSS PRODUCT QT, DIVIDED BY R**2, THE
C*                     SQUARE OF THE RADIUS OF THE EARTH.  SOLVING THIS
C*                     EQUATION FOR THETA, THE COLATITUDE, LEADS TO A
C*                     DIRECTLY SOLVABLE EQUATION,
C*
C*                       THETA = ATAN(-QTZ/(QTX*COS(PHI) + QTY*SIN(PHI))
C*
C*                     ON THE OTHER HAND, SOLVING FOR PHI LEADS TO A
C*                     TRANSCENDENTAL EQUATION OF THE FORM,
C*
C*                       QTX*COS(PHI) + QTY*SIN(PHI) + QTZ/TAN(THETA) = 0
C*
C*                     THIS LATTER EQUATION IS SOLVED BY NEWTON'S ITERA-
C*                     TIVE TECHNIQUE FOR NON-LINEAR EQUATIONS.  THE
C*                     CRITICAL FIRST GUESS FOR PHI IS PROVIDED BY ONE
C*                     OF THREE METHODS.  DEPENDING ON USER DESIRES AND
C*                     WHETHER THE POINT A-PRIM HAS ALREADY BEEN ESTAB-
C*                     LISHED, ANY ONE OF THE THREE FIRST GUESS METHODS
C*                     CAN BE USED.  THE METHODS ARE...
C*
C*                       (1)  USER-SPECIFIED FIRST GUESS
C*
C*                       (2)  APPROXIMATELY PLANAR TRIANGLE
C*
C*                       (3)  EXACT SPHERICAL TRIANGLE.
C*
C*                     NORMALLY THE THIRD METHOD IS USED, BUT METHOD (2)
C*                     MUST BE USED AT LEAST ONCE TO BREAK GROUND FOR
C*                     (3).  METHOD (2) IS ONLY APPROXIMATE AND LOSES
C*                     ACCURACY THE LONGER THE DIS-
C*                     TANCE BETWEEN 'A' AND 'B.'  THIS CAN BE A MAJOR
C*                     LIMITING FACTOR, BECAUSE NEWTON'S TECHNIQUE REQUIRES
C*                     A REASONABLY CLOSE FIRST GUESS FOR PHI.
C*
C*   LIMITATIONS-      1.  THE APPROXIMATELY PLANAR TRIANGLE
C*                     METHOD USED AS FIRST GUESS FOR NEWTON'S ITERATIVE
C*                     TECHNIQUE APPEARS TO BREAK DOWN WHEN ABSOLUTE VAL-
C*                     OF THE DELTA-LONGITUDE OR DELTA-LATITUDE BETWEEN
C*                     POINTS 'A' AND 'B' EXCEEDS ABOUT 40-50 DEGREES, PAR-
C*                     TICULARLY WHEN THE TWO POINTS ARE IN DIFFERENT QUAD-
C*                     RANTS OF THE GLOBE.  UNDER THESE CIRCUMSTANCES, NEW-
C*                     TON'S METHOD CONVERGES TO A FALSE ROOT OR FAILS TO
C*                     CONVERGE ENTIRELY, AND IT BECOMES IMPOSSIBLE TO
C*                     SOLVE THE EQUATION OF A GREAT CIRCLE FOR THE AZIMUTH
C*                     ANGLE PHI.  POINTS 'A' AND 'B' ALONG THE GREAT
C*                     CIRCLE MUST THUS BE KEPT REASONABLY CLOSE TOGETHER.
C*                     IF POINT 'B' IS TOO FAR REMOVED FROM 'A,' THEN USE
C*                     THIS SUBROUTINE DETERMINISTICALLY (ICONV = 1 AND 2
C*                     FOLLOWED BY ICONV = 3 OR 4) TO CALCULATE THETA
C*                     (XLAT) FOR AN INTERMEDIATE PHI (XLNG) NOT MORE THAN
C*                     30 DEGREES FROM PHIA.  THEN USE THAT POINT AS THE
```

```
C*                        NEW POINT B IN ALL FURTHER GREAT CIRCLE COMPUTA-
C*                        TIONS.  THIS CAN BE DONE BECAUSE THE 'INTERMEDIATE'
C*                        POINT NEED NOT BE BOUNDED BY 'A' AND 'B.'
C*
C*                        2.  IN THE CASE OF A GREAT CIRCLE THAT IS ALSO
C*                        A LONGITUDE CIRCLE, THE IMPLICIT FUNCTION
C*                        F(THETA,PHI) = 0.0 DOES NOT RETURN A UNIQUE
C*                        SOLUTION IN THETA FOR A SPECIFIED PHI.  IF THE
C*                        GREAT CIRCLE IS ALONG A LONGITUDE, ANY VALUE
C*                        OF THETA IS A SOLUTION TO THE FUNCTION.  THIS
C*                        SUBROUTINE CANNOT BE USED TO CALCULATE THETA
C*                        FROM PHI IF FRMLNG = TOLNG.  AN ERROR MESSAGE
C*                        IS DISPLAYED ON SYSOUT AND ERROR FLAG ICONV=12
C*                        IS SET IF THIS IS ATTEMPTED.
C*
C*   INPUT-               FRMLAT = LATITUDE OF POINT 'A' (ORIGIN) IN DEG
C*
C*                        FRMLNG = LONGITUDE OF POINT 'A' (ORIGIN) IN DEG
C*
C*                        TOLAT  = LATITUDE OF POINT 'B' (DESTINATION) IN DEG
C*
C*                        TOLNG  = LONGITUDE OF POINT 'B' (DESTINATION) IN DEG
C
C*                        THETAA = COLATITUDE ANGLE IN RADIANS OF POINT 'A,'
C*                                 0.0 TO PI RADIANS SOUTHWARD FROM NORTH,
C*                                 POSITIVE
C*
C*                        PHIA   = AZIMUTH ANGLE IN RADIANS COUNTERCLOCKWISE
C*                                 FROM GREENWICH LOOKING DOWNWARD AT THE
C*                                 NORTH POLE.  PLUS OR MINUS 0.0 TO PI RADIAN
C*                                 (NEGATIVE MEANS CLOCKWISE)
C*
C*                        THETAB = COLATITUDE IN RADIANS OF POINT 'B'
C*
C*                        PHIB   = AZIMUTH IN RADIANS OF POINT 'B'
C*
C*                        IGUESS = FLAG CONTROLLING WHETHER FIRST GUESS IS
C*                                 SPECIFIED FOR LONGITUDE IN VARIABLE
C*                                 XLNGG.   IF IGUESS = 1, XLNGG IS INTER-
C*                                 PRETED AS FIRST GUESS LONGITUDE.  IF
C*                                 IGUESS = 0, XLNGG IS IGNORED ON INPUT
C*                                 BUT IS FILLED WITH THE PROGRAM-GENERATED
C*                                 FIRST GUESS VALUE ON OUTPUT.
C*
C*                        XEAST  = EASTWARD DIRECTION INDICATOR, POSITIVE
C*                                 FOR AIRCRAFT MOTION TOWARD THE EAST.
C*
C*                        XNORTH = NORTHWARD DIRECTION INDICATOR, POSITIVE
C*                                 FOR AIRCRAFT MOTION TOWARD THE NORTH.
C*
C*                        XLATC  = LATITUDE OF CURRENT-POINT, I.E., THE
C*                                 POINT FROM WHICH THE PROGRAM IS MARCHING
C*                                 IN OBTAINING XLAT, THE LATITUDE OF THE
C*                                 SO CALLED NEXT-POINT.
C*
C*   INPUT &
C*   OUTPUT-              ICONV  = FLAG CONTROLLING FUNCTION TO BE PERFORMED
C*                                 BY THIS SUBROUTINE.  SEE TABLE IN 'METHOD'
C*                                 ABOVE.  WHEN RETURNED AS OUTPUT, ICONV = 1
C*                                 THROUGH 6 FOR NORMAL TERMINATION OR A NUM-
C*                                 BER GREATER THAN OR EQUAL TO 10 FOR ABNOR-
C*                                 MAL TERMINATION.  ABNORMAL TERMINATION
C*                                 CODES FOR ICONV ARE...
C*
C*                                    10-  ERRONEOUS ICONV SPECIFIED
C*                                    11-  NEWTON'S ITERATIVE TECHNIQUE FOR
C*                                         SOLUTION OF NON-LINEAR EQUATIONS
C*                                         DID NOT CONVERGE IN NMAX ITERA-
```

```
C*                                   TIONS
C*                              12-  ATTEMPT WAS MADE TO CALCULATE
C*                                   THETA FROM PHI FOR A FLIGHT
C*                                   ALONG A LONGITUDE CIRCLE
C*
C*                 XLAT   = LATITUDE OF INTERMEDIATE POINT IN DEG
C*
C*                 XLNG   = LONGITUDE OF INTERMEDIATE POINT IN DEG
C*
C*                 QTX    = X-COMPONENT OF CROSS PRODUCT QT DIVIDED BY
C*                          R**2.  COMPUTED IN FUNCTION ICONV = 1 AND
C*                          REUSED IN SUBSEQUENT CALLS
C*
C*                 QTY    = Y-COMPONENT OF ABOVE
C*
C*                 QTZ    = Z-COMPONENT OF ABOVE
C*
C*                 XLNGG  = FIRST GUESS LONGITUDE FOR ICONV = 4
C*                          OR 5.  IF VALUE IS NOT SPECIFIED AS
C*                          INPUT, THE PROGRAM-GENERATED FIRST
C*                          GUESS WILL BE PROVIDED AS OUTPUT
C*
C*
C*                 XLNGAP = LONGITUDE IN DEGREES OF THE POINT A-PRIME
C*                          AT WHICH THE GREAT CIRCLE CROSSES THE
C*                          EQUATOR
C*
C*                 SAPRIM = SINE OF INTERIOR ANGLE A-PRIME OF THE
C*                          EXACT SPHERICAL TRIANGLE
C*
C*                 THETA  = COLATITUDE OF INTERMEDIATE POINT IN RADIANS
C*
C*                 PHI    = AZIMUTH OF INTERMEDIATE POINT IN RADIANS
C*
C*                 NOTE-  ALL SOUTH LATITUDES AND EAST LONGITUDES ARE
C*                 REPRESENTED AS NEGATIVE NUMBERS.
C*
C*   SYSTEM SUB-
C*   PROGRAMS
C*   USED-         SIN, ARSIN, COS, ARCOS, TAN, ATAN, FXEM
C*
C*   USER SUB-
C*   PROGRAMS
C*   USED-         SPHGLO, HDG
C*
C*   ESTIMATED
C*   CPU TIME-     TO BE SUPPLIED.
C*
C*   STORAGE
C*   REQUIREMENTS-  PROCEDURE PLUS DATA OCCUPY 1042 WORDS OF STORAGE.
C*
C*   PROGRAM
C*   UPDATES-      NONE
C*
C***********************************************************************
C
C     EPSILON IS THE CRITERION FOR CONVERGENCE OF NEWTON'S ITERATIVE
C      TECHNIQUE AND HAS UNITS OF RADIANS.  PI IS THE GEOMETRIC PI.
C      NMAX IS THE MAXIMUM NUMBER OF ITERATIONS PERMITTED FOR THE NEW-
C      TON TECHNIQUE.  NTRIFG IS A FLAG INDICATING WHETHER THE
C      DIRECTION (TRIDIR) OF THE SPHERICAL TRIANGLE IN FIRST
C      GUESS METHOD #3 HAS BEEN REVERSED DUE TO AN EQUATOR
C      CROSSING.  NORMALLY ZERO, NTRIFG = 1 INDICATES A REVERSAL.
C
C     COMMON /DBG/ NDEBUG
      DIMENSION APRMD(2), XLNGGG(2), PHIGG(2)
      DATA      EPS/0.0001/, PI/3.1415927/, NTRIFG/0/, NMAX/7/
C
```

```
C     BRANCH TO APPROPRIATE SECTION OF THE SUBROUTINE BASED ON THE FUNC-
C      TION REQUESTED VIA THE ICONV FLAG.  THE ERROR CONDITION ICONV=10
C      IS RETURNED FOR ILLEGAL ICONV VALUES.  IF FLIGHT IS ALONG
C      A LONGITUDE CIRCLE AND ICONV REQUESTS CALCULATION OF THETA,
C      AN ERROR CONDITION ICONV=12 IS SET AND AN ERROR MESSAGE IS
C      DISPLAYED ON SYSOUT.  ICONVS IS A SAVED VALUE OF THE ORIGINAL
C      ICONV.
C
      ICONVS = ICONV
      IF (ICONV .GE. 1 .AND. ICONV .LE. 5) GO TO 50
      ICONV = 10
      GO TO 1000
   50 IF (ICONV .NE. 2 .AND. ICONV .NE. 3) GO TO 60
      IF (ABS(FRMLNG - TOLNG) .GT. 0.0001) GO TO 60
      ICONV = 12
      GO TO 1020
   60 GO TO (100, 200, 300, 400, 500, 600), ICONV
C
C
C                 **************************************
C                 *              PART I                *
C                 *         FUNCTION ICONV = 1         *
C                 *     CROSS PRODUCT COMPUTATION      *
C                 **************************************
C
C     CONVERT (LAT,LNG) OF 'A' AND 'B' TO (THETAA,PHIA) AND (THETAB,
C      PHIB).
C
  100 ICONVX = 1
      CALL SPHGLO (ICONVX, FRMLAT, FRMLNG, THETAA, PHIA)
      CALL SPHGLO (ICONVX,  TOLAT,  TOLNG, THETAB, PHIB)
C
C     PRELIMINARIES FOR COMPONENTS OF POSITION VECTORS QA AND QB.
C
      SINTHA = SIN(THETAA)
      COSTHA = COS(THETAA)
      SINPHA = SIN(PHIA  )
      COSPHA = COS(PHIA  )
      SINTHB = SIN(THETAB)
      COSTHB = COS(THETAB)
      SINPHB = SIN(PHIB  )
      COSPHB = COS(PHIB  )
C
C     RECTANGULAR COMPONENTS OF POSITION VECTORS QA AND QB, DIVIDED
C      BY THE RADIUS OF THE EARTH R.
C
      QAX = SINTHA * COSPHA
      QAY = SINTHA * SINPHA
      QAZ = COSTHA
      QBX = SINTHB * COSPHB
      QBY = SINTHB * SINPHB
      QBZ = COSTHB
C
C     COMPUTE COMPONENTS OF THE CROSS PRODUCT VECTOR QT = QA X QB,
C      DIVIDED BY THE RADIUS OF THE EARTH SQUARED, R**2.
C
      QTX  =  QAY * QBZ  -  QAZ * QBY
      QTY  =  QAZ * QBX  -  QAX * QBZ
      QTZ  =  QAX * QBY  -  QAY * QBX
C
C     INITIALIZE TRIANGLE FLAG, USED TO REVERSE THE DIRECTION
C      TRIDIR OF THE SPHERICAL TRIANGLE IN THE FIRST GUESS PRO-
C      CEDURE WHENEVER THE ROUTE CROSSES THE EQUATOR.
C
      NTRIFG = 0
C
C     END OF FUNCTION ICONV = 1.  RETURN TO MAIN PROGRAM.
C
```

```
      RETURN
C
C
C                 ************************************
C                 *             PART II              *
C                 *        FUNCTION ICONV = 2        *
C                 *     EQ CROSSING PT $ ANGLE       *
C                 ************************************
C
C     DETERMINE WHETHER POINT 'A' OR POINT 'B' IS TO SERVE AS POINT
C      'E.'  WHICHEVER IS CLOSEST TO THE EQUATOR WILL SERVE AS 'E.'
C      THE OTHER WILL SERVE AS 'F.'  NSIGN IS A FLAG INDICATING THE
C      ROUTE INVOLVES AN EQUATOR CROSSING.  NORMALLY 1, NSIGN TAKES
C      ON THE VALUE 2 FOR ROUTES CROSSING THE EQUATOR.  NCASE IS
C      A FLAG THAT HAS THE VALUE 12 WHEN POINT 'A' SERVES AS POINT
C      'E' AND THE VALUE 34 WHEN POINT 'B' SERVES AS POINT 'E.'
C
  200 NSIGN  =  1
      IF (FRMLAT .LT. 0.0 .AND. TOLAT .GT. 0.0) NSIGN = 2
      IF (FRMLAT .GT. 0.0 .AND. TOLAT .LT. 0.0) NSIGN = 2
      XLATE  =  FRMLAT
      XLNGE  =  FRMLNG
      PHIE   =  PHIA
      XLATF  =  TOLAT
      XLNGF  =  TOLNG
      PHIF   =  PHIB
      NCASE  =  12
      IF ( ABS(XLATE) .LT. ABS(XLATF) )  GO TO 205
      XLATE  =  TOLAT
      XLNGE  =  TOLNG
      PHIE   =  PHIB
      XLATF  =  FRMLAT
      XLNGF  =  FRMLNG
      PHIF   =  PHIA
      NCASE  =  34
  205 CONTINUE
C     IF (NDEBUG .GT. 1) WRITE (6,6000) NSIGN, NCASE
C6000 FORMAT (1H , T4, 'NSIGN,NCASE = ', T36, 2I10)
C
C     ESTABLISH INITIAL HEADING OF AIRCRAFT AT POINT 'E' ON THE GREAT
C      CIRCLE.  CONVERT HEADING AT 'E' IN DEGREES TO ANGLE 'E' IN
C      RADIANS.  TAKE ITS TANGENT TANE FOR LATER USE IN THE FIRST
C      GUESS PROCEDURE #2.
C
      HDGE = HDG (XLATE, XLNGE, XLATF, XLNGF)
      IF (HDGE .GE. 270.0) GO TO 212
      IF (HDGE .GE. 180.0) GO TO 214
      IF (HDGE .GE.  90.0) GO TO 216
      GO TO 218
  212 E = HDGE - 270.0
      GO TO 220
  214 E = 270.0 - HDGE
      GO TO 220
  216 E = HDGE -  90.0
      GO TO 220
  218 E =  90.0 - HDGE
      GO TO 220
  220 TANE = TAN (E * (PI/180.0))
C     IF (NDEBUG .GT. 1) WRITE (6,6010) HDGE, E
C6010 FORMAT (1H , T4, 'HEADING AT POINT E = ', T36, F10.4,
C      ' DEG'/1X, T4, T4, 'ANGLE E = ', T36, F10.4, ' DEG')
C
C     SET LATITUDE TO ZERO AND DETERMINE LONGITUDE WHERE GREAT CIRCLE
C      CROSSES EQUATOR.  THIS IS POINT A-PRIME, WHOSE LONGITUDE IS
C      XLNGAP IN DEGREES.
C
      XLAT   = 0.0
      GO TO 500
```

```
  230 XLNGAP = XLNG
C     IF (NDEBUG .GT. 1) WRITE (6,6020) XLNGAP
C6020 FORMAT (1H , T4, 'LONGITUDE OF EQUATOR CROSSING = ', T36, F10.4,
C      ' DEG')
C
C     IF LONGITUDE XLNGAP OF EQUATOR CROSSING POINT 'A-PRIME' IS
C      WEST OF LONGITUDE OF POINT 'A' (FRMLNG), THE SPHERICAL
C      TRIANGLE OPENS EASTWARD (TRIDIR = -1.0).  OTHERWISE, IT
C      OPENS WESTWARD (TRIDIR = 1.0).
C
      XLGAPP = XLNGAP
      FRMLNN = FRMLNG
      IF (XLGAPP .LT. -90.0) XLGAPP = XLGAPP + 360.0
      IF (FRMLNN .LT. -90.0) FRMLNN = FRMLNN + 360.0
      IF (FRMLNN - XLGAPP  ) 240,245,245
  240 TRIDIR = -1.0
      GO TO 250
  245 TRIDIR =  1.0
  250 CONTINUE
C     IF (NDEBUG .GT. 1) WRITE (6,6030) TRIDIR
C6030 FORMAT (1H , T4, 'TRIDIR = ', T36, F10.4)
C
C     ESTABLISH INITIAL HEADING OF THE AIRCRAFT AT POINT A-PRIME ON
C      CIRCLE.  CONVERT HEADING AT A-PRIME TO ANGLE A-PRIME.  CONVERT
C      ANGLE A-PRIME TO RADIANS AND TAKE ITS SINE FOR LATER USE IN
C      FIRST GUESS PROCEDURE #3.
C
  260 HDGAP = HDG (XLAT, XLNGAP, TOLAT, TOLNG)
      IF (HDGAP .GE. 270.0) GO TO 270
      IF (HDGAP .GE. 180.0) GO TO 275
      IF (HDGAP .GE.  90.0) GO TO 280
      GO TO 285
  270 APRIMD = HDGAP - 270.0
      GO TO 290
  275 APRIMD = 270.0 - HDGAP
      GO TO 290
  280 APRIMD = HDGAP -  90.0
      GO TO 290
  285 APRIMD =  90.0 - HDGAP
      GO TO 290
  290 SAPRIM = SIN (APRIMD * (PI/180.0))
C     IF (NDEBUG .GT. 1) WRITE (6,6040) APRIMD
C6040 FORMAT (1H , T4, 'APRIMD = ', T36, F10.4, ' DEG')
C
C     END OF FUNCTION ICONV = 2.  RETURN TO MAIN PROGRAM.
C
      RETURN
C
C
C                    *************************************
C                    *             PART III              *
C                    *         FUNCTION ICONV = 3        *
C                    *           XLNG TO XLAT            *
C                    *************************************
C
C     CONVERT INTERMEDIATE LONGITUDE XLNG TO INTERMEDIATE PHI.
C
  300 ICONVX = 1
      CALL SPHGLO (ICONVX, 0.0, XLNG, THETAX, PHI)
C
C
C                    *************************************
C                    *             PART IV               *
C                    *         FUNCTION ICONV = 4        *
C                    *            PHI TO XLAT            *
C                    *************************************
C
C     CALCULATE THETA FROM PHI.
```

```
      C
        400 THETA  =  ATAN(-QTZ / (QTX * COS(PHI)  +  QTY * SIN(PHI) ) )
            IF (THETA) 410,900,900
        410 THETA  =  THETA  +  PI
            GO TO 900
      C
      C
      C                    **************************************
      C                    *              PART V                *
      C                    *         FUNCTION ICONV = 5         *
      C                    *            XLAT TO XLNG            *
      C                    **************************************
      C
      C     SAVE AZIMUTH ANGLE OF THE CURRENT POINT IN PHIC FOR LATER USE IN
      C      CHECKING FOR MONOTONIC PROGRESS IN LONGITUDE.
      C
        500 PHIC = PHI
      C
      C     CONVERT INTERMEDIATE LATITUDE XLAT TO INTERMEDIATE THETA.
      C
            ICONVX = 1
            CALL SPHGLO (ICONVX, XLAT, 0.0, THETA, PHIX)
      C
      C
      C                    **************************************
      C                    *              PART VI               *
      C                    *         FUNCTION ICONV = 6         *
      C                    *           THETA TO XLNG            *
      C                    **************************************
      C
      C     CALCULATE PHI FROM THETA (NEWTON'S ITERATIVE TECHNIQUE WITH AN
      C      APPROPRIATE FIRST GUESS).
      C
      C     THREE METHODS CAN BE USED AS A FIRST GUESS FOR THE AZIMUTH
      C      ANGLE PHI.  THE FIRST AND LEAST SOPHISTICATED METHOD IS
      C      FOR THE USER TO SPECIFY A FIRST GUESS BY SETTING INPUT
      C      IGUESS = 1 AND BY INSERTING A FIRST GUESS LONGITUDE IN
      C      XLNGG IN HIS CALL TO THE GRTCIR SUBROUTINE.  THIS METHOD
      C      IS IMPLEMENTED IN STATEMENTS 605-609.
      C
      C     THE SECOND FIRST GUESS METHOD IS THAT OF APPROXIMATELY
      C      PLANAR TRIANGLES.  THIS METHOD HAS IMPORTANT LIMI-
      C      TATIONS ON ITS RELIABILITY AND SO IS USED ONLY TO FIND THE
      C      EQUATORIAL LONGITUDE OF THE GREAT CIRCLE (FUNCTION ICONV =
      C      2).  THE APPROXIMATELY PLANAR TRIANGLE METHOD
      C      IS IMPLEMENTED IN STATEMENTS 610-624.
      C
      C     THE THIRD AND MOST SOPHISTICATED FIRST GUESS METHOD IS THAT
      C      OF EXACT SPHERICAL TRIANGLES.   HIS METHOD REQUIRES
      C      KNOWLEDGE OF THE EQUATORIAL C    ING POINT AND ANGLE.
      C      THE EXACT SPHERICAL TRIANGLE     .T GUESAS METHOD IS
      C      IMPLEMENTED IN STATEMENTS 625-700.
      C      625-700.
      C
        600 IF (IGUESS .EQ. 1) GO TO 605
            IF (ICONV  .EQ. 2) GO TO 610
            GO TO 625
      C
      C
      C                  *****FIRST GUESS METHOD #1*****
      C
      C     SPECIFIED FIRST GUESS.  STATEMENTS 605-609.
      C
        605 ICONVX  = 1
            DUMLAT  = 0.0
            DUMTHE  = 0.0
            CALL SPHGLO (ICONVX, DUMLAT, XLNGG, DUMTHE, PHIG)
            PHIN    = PHIG
```

```
      ICONVX  = 2
      CALL SPHGLO (ICONVX, DUMLAT, XLNGG, DUMTHE, PHIG)
  609 GO TO 750
C
C
C                   *****FIRST GUESS METHOD #2*****
C
C     APPROXIMATELY PLANAR TRIANGLE FIRST GUESS.  STATE-
C      MENTS 610-624.
C
  610 IF (NSIGN .EQ. 1) GO TO 612
      IF (NSIGN .EQ. 2) GO TO 616
  612 PHIN  =  PHIE  +  ABS( XLATE * (PI/180.0) * TANE )
      PHIG  =  PHIN
      GO TO 620
  616 PHIN  =  PHIE  +  ABS( XLATE * (PI/180.0) / TANE )
      PHIG  =  PHIN
      GO TO 620
  620 ICONVX = 2
      CALL SPHGLO (ICONVX, DUMLAT, XLNGG, DUMTHE, PHIG)
C     IF (NDEBUG .GT. 1) WRITE (6,6050) PHIE, XLATE, PHIN
C6050 FORMAT (1H , T4, 'PHIE,XLATE,PHIN = ', T36, 3F10.4,
C      ' RAD,DEG,RAD')
C     IF (NDEBUG .GT. 1) WRITE (6,6060) XLNGG, PHIG
C6060 FORMAT (1H , T4, 'FIRST GUESS LONGITUDE & AZIMUTH = ',
C      T36, 2F10.4, ' DEG,RAD')
  624 GO TO 750
C
C
C                   *****FIRST GUESS METHOD #3*****
C
C     EXACT SPHERICAL TRIANGLE FIRST GUESS.  STATEMENTS 625-700.
C
C     LATITUDE XLAT OF THE INTERMEDIATE POINT GIVES THE ANGULAR
C      DISTANCE DE-PRIME IN RADIANS.
C
  625 DEPRIM = XLAT * (PI/180.0)
C     IF (NDEBUG .GT. 1) WRITE (6,6070) DEPRIM
C6070 FORMAT (1H , T4, 'DEPRIM = ', T36, F10.4, ' RAD')
C
C     LAW OF SINES FOR SPHERICAL TRIANGLES GIVES SOLUTION FOR ANGULAR
C      DISTANCE APRMD(1) AND APRMD(2) IN RADIANS.  APRMD(2) IS
C      (180.0-APRMD(1)) AND HAS THE SAME SINE AS APRMD(1).  THIS PRO-
C      CEDURE IS NECESSARY BECAUSE THE ARC SINE FUNCTION RETURNS
C      ONLY THE PRINCIPAL ANGLE.
C
      SRATIO = SIN(DEPRIM) / SAPRIM
C
C     IF THE ABSOLUTE VALUE OF SRATIO EXCEEDS THE LEGAL RANGE, THEN
C      THE SOLUTION IN LONGITUDE IS INDETERMINATE FOR A GIVEN LATI-
C      TUDE.  IN SUCH A CASE, THE METHOD MAY BE SAID TO HAVE FAILED
C      TO CONVERGE, AND ERROR (ICONV = 11) IS ASSIGNED.
C
      IF ( ABS(SRATIO) .LE. 1.0) GO TO 626
      ICONV = 11
      GO TO 1010
  626 APRMD(1) = ARSIN ( SRATIO )
      APRMD(2) = PI - APRMD(1)
C     IF (NDEBUG .GT. 1) WRITE (6,6080) APRMD(1), APRMD(2)
C6080 FORMAT (1H , T4, 'APRMD(1),APRMD(2) = ', T36, 2F10.4, ' RAD')
C
C     INVERSE OF FORMULA FOR GREAT CIRCLE DISTANCE (SEE SUBROUTINE
C      'DISTAN') GIVES FIRST GUESSES XLNGGG(I), I = 1,2, FOR LATITUDE
C      OF INTERMEDIATE POINT IN DEGREES.  THE FACT THAT THERE ARE
C      TWO XLNGGG VALUES EMERGES FROM THE FACT THAT THERE ARE TWO
C      APRMD VALUES.  SIGN MULTIPLIER TRIDIR (-1.0 OR 1.0) CAUSES
C      ADDITION OF LONGITUDE ELEMENT FOR THE CASE WHERE POINT
C      'A-PRIME' IS EAST OF POINT 'A' (SPHERICAL TRIANGLE OPENS
```

```
C        WESTWARD).  OTHERWISE, LONGITUDE ELEMENT IS SUBTRACTED.
C
C       SPECIAL CODE IS NEEDED FOR THE CASE OF AN EQUATOR CROSSING.
C        THE CODE IMMEDIATELY FOLLOWING AND EXTENDING THROUGH STATE-
C        MENT 629 ACTS TO REVERSE THE DIRECTION OF THE SPHERICAL
C        TRIANGLE AT THE POINT WHERE AN AIRCRAFT PATH CROSSES THE
C        EQUATOR SOUTHWARD FROM THE NORTHERN HEMISPHERE OR NORTHWARD
C        FROM THE SOUTHERN HEMISPHERE.  ONLY IF NSIGN EQUALS 2 WILL
C        AN EQUATOR CROSSING OCCUR.  AN EQUATOR CROSSING IS DECLARED
C        WHEN THE CURRENT POINT XLATC IS 0.0 AND THE NEXT-POINT IS
C        NEGATIVE (FOR A SOUTHWARD CROSSING) OR POSITIVE (FOR A
C        NORTHWARD CROSSING).  WHETHER A FLIGHT IS PROCEEDING NORTH-
C        WARD OR SOUTHWARD IS INDICATED BY THE XNORTH FLAG.
C
        IF (NTRIFG .EQ. 1) GO TO 629
        IF (NSIGN .EQ. 1) GO TO 629
        IF (XNORTH) 627,628,628
    627 IF (XLATC .NE. 0.0) GO TO 629
        IF (XLAT  .GE. 0.0) GO TO 629
        TRIDIR = -TRIDIR
        NTRIFG = 1
        GO TO 629
    628 IF (XLATC .NE. 0.0) GO TO 629
        IF (XLAT  .LE. 0.0) GO TO 629
        TRIDIR = -TRIDIR
        NTRIFG = 1
    629 CONTINUE
C       IF (NDEBUG .GT. 1) WRITE (6,6090) TRIDIR
C6090 FORMAT (1H , T4, 'TRIDIR = ', T36, F10.4)
C
        DO 640 I = 1,2
    630 XLNGGG(I)  = XLNGAP  +  TRIDIR * ( (180.0/PI) *
     &   ARCOS ( COS(APRMD(I)) / COS( XLAT * (PI/180.0) ) ) )
C
C       ADJUST COMPUTED LONGITUDE XLNGG FOR DATELINE FOLD.
C
        IF (XLNGGG(I) .GT.  180.0) GO TO 632
        IF (XLNGGG(I) .LT. -180.0) GO TO 634
        GO TO 636
    632 XLNGGG(I) = XLNGGG(I) - 360.0
        GO TO 636
    634 XLNGGG(I) = XLNGGG(I) + 360.0
        GO TO 636
C
C       CONVERT LONGITUDES XLNGGG(I) IN DEGREES TO AZIMUTHS PHIGG(I) IN
C        RADIANS.
C
    636 ICONVX = 1
        CALL SPHGLO (ICONVX, DUMLAT, XLNGGG(I), DUMTHE, PHIGG(I))
C       IF (NDEBUG .GT. 1) WRITE (6,6100) XLNGGG(I)
C6100 FORMAT (1H , T4, 'XLNGGG(I) = ', T36, F10.4, ' DEG')
    640 CONTINUE
C
C       INITIALLY, WHICHEVER LONGITUDE XLNGGG(I) HAS AN ASSOCIATED
C        AZIMUTH PHIGG(I) ABSOLUTELY CLOSER TO PHIC, THE AZIMUTH
C        ANGLE OF THE CURRENT POINT, IS SELECTED AS THE FIRST GUESS
C        LONGITUDE XLNGG IN DEGREES WITH CORRESPONDING AZIMUTH PHIG
C        (POINTER IS SAVED IN IGOOD).  IF XEAST IS GREATER THAN
C        ZERO (LESS THAN ZERO), PHIG MUST BE GREATER THAN (LESS
C        THAN) PHIC.  IF THIS IS NOT THE CASE, THE ALTERNATE LONGI-
C        TUDE SOLUTION IS USED AS FIRST GUESS (POINTERS IGOOD AND
C        IBAD ARE SWAPPED).
C
C       IF (NDEBUG .GT. 1) WRITE (6,6110) PHIGG, PHIC, XEAST
C6110 FORMAT (1H , T4, 'PHIG(1)-AND-(2),PHIC,XEAST = ', T36, 4F10.4)
        IGOOD = 1
        IBAD  = 2
        IF ( (ABS(PHIGG(IBAD) - PHIC)) .GT. (ABS(PHIGG(IGOOD) - PHIC)) )
```

```
     &       GO TO 645
             ITEMP = IGOOD
             IGOOD = IBAD
             IBAD  = ITEMP
   645 IF (PHIGG(IGOOD) - PHIC) 650,652,652
   650 IF (XEAST)                660,654,654
   652 IF (XEAST)                654,654,660
   654 ITEMP = IGOOD
       IGOOD = IBAD
       IBAD  = ITEMP
   660 CONTINUE
C      IF (NDEBUG .GT. 1) WRITE (6,6120) XLNGGG(IGOOD), PHIGG(IGOOD),
C       XLNGGG(IBAD), PHIGG(IBAD)
C6120 FORMAT (1H , T4, 'XLNGGG,PHIGG FOR IGOOD = ', T36, 2F10.4/1X,
C       T4, 'XLNGGG,PHIGG FOR IBAD  = ', T36, 2F10.4)
C
C      SET FIRST GUESS LONGITUDE AND PHI.
C
       XLNGG = XLNGGG(IGOOD)
       PHIG  = PHIGG (IGOOD)
C
C      STORE PHIG IN PHIN.
C
   690 PHIN = PHIG
   700 GO TO 750
C
C
C                    *****NEWTON'S ITERATIVE TECHNIQUE*****
C
C      IMPROVE BY NEWTON'S METHOD.
C
C      INITIALIZE ITERATION COUNT AND DENOMINATOR.
C
   750 N = 0
       NP1 = N + 1
       TANTHE = TAN(THETA)
C
C      FUNCTION.
C
   775 FPHI  =  QTX * COS(PHIN)  +  QTY * SIN(PHIN)  +  QTZ / TANTHE
C
C      FIRST DERIVATIVE OF FUNCTION.
C
       FPPHI = -QTX * SIN(PHIN)  +  QTY * COS(PHIN)
C
C      IMPROVED PHI.
C
       PHINP1  =  PHIN  -  (FPHI / FPPHI)
C
C      TEST FOR CONVERGENCE.  STOP WITH ERROR CONDITION ICONV = 11 IF
C       METHOD FAILS TO CONVERGE IN NMAX ITERATIONS.
C
       DELPHI  =  ABS(PHINP1 - PHIN)
       IF (DELPHI - EPS) 850,850,800
   800 PHIN = PHINP1
       N = N + 1
       NP1 = N + 1
       IF (N .LT. NMAX) GO TO 775
       ICONV = 11
       GO TO 1010
   850 PHI = PHINP1
C
```

```
C
C                   ************************************
C                   *            PART VII              *
C                   *          ALL FUNCTIONS           *
C                   *   CONVERT SOLUTION (THETA,PHI)   *
C                   *    BACK TO GLOBAL (XLAT,XLNG)    *
C                   ************************************
C
  900 ICONVX = 2
      CALL SPHGLO (ICONVX, XLAT, XLNG, THETA, PHI)
      IF (ICONV .EQ. 2) GO TO 230
C
C
C                   ************************************
C                   *            PART VIII             *
C                   *        NORMAL TERMINATION        *
C                   ************************************
C
      RETURN
C
C
C                   ************************************
C                   *             PART IX              *
C                   *       ABNORMAL TERMINATION       *
C                   ************************************
C
 1000 WRITE (6,6130) FRMLAT, FRMLNG, TOLAT, TOLNG
 6130 FORMAT (1HO, '****WIND FACTOR ERROR DIAGNOSTICS****'/1X,
&       'FROM-LAT     = ', F10.2, T30, 'FROM-LNG = ', T41, F10.2/1X,
&       'TO-LAT       = ', F10.2, T32, 'TO-LNG = ', T41, F10.2//)
      CALL FXEM (61, 'SUBROUTINE GRTCIR...ILLEGAL ICONV', 9)
      GO TO 1050
 1010 CONTINUE
      GO TO 1050
 1020 WRITE (6,6130) FRMLAT, FRMLNG, TOLAT, TOLNG
      CALL FXEM (61, 'SUBROUTINE GRTCIR...SOLUTION NOT UNIQUE IN THETA',
&       12)
 1050 IF (ICONVS .EQ. 2 .OR. ICONVS .EQ. 3) GO TO 1060
      IF (ICONVS .EQ. 4 .OR. ICONVS .EQ. 5) GO TO 1070
 1060 XLAT  = 0.0
      THETA = 0.0
      RETURN
 1070 XLNG  = 0.0
      PHI   = 0.0
      RETURN
      END
```

```
CSPHGLO CONV SPH<>GLOBAL/R. C. WHITON/05 FEB 1979
C
      SUBROUTINE SPHGLO (ICONV, XLAT, XLON, THETA, PHI)
C
C************************************************************************
C*
C*   PROGRAM ID-     SPHGLO
C*   MET ANALYST-    MAJ ROGER C. WHITON, USAFETAC/DNS, EXT 5412
C*   SYS ANALYST-    MAJ ROGER C. WHITON, USAFETAC/DNS, EXT 5412
C*   PROGRAMMER-     MAJ ROGER C. WHITON, USAFETAC/DNS, EXT 5412
C*
C*   CREATED ON-     05 FEB 1979            PROJECT-  192301
C*
C*   DESCRIPTION-    THIS SUBROUTINE SUBPROGRAM CONVERTS GLOBAL COOR-
C*                   DINATES (LAT,LON) TO MATHEMATICAL SPHERICAL COOR-
C*                   DINATES (THETA,PHI), OR VICE-VERSA, DEPENDING ON
C*                   THE VALUE OF THE ICONV FLAG (ICONV=1 FOR CONVER-
C*                   SION OF (LAT,LON) TO (THETA,PHI), AND ICONV=2 FOR
C*                   CONVERSION OF (THETA,PHI) TO (LAT,LON).
C*
C*   METHOD-         FOR CONVERSION OF (LAT,LON) TO (THETA,PHI)...
C*
C*                      THETA = (PI/2.0) - XLAT
C*                      PHI   = -XLON              FOR XLON .LE. 0.0
C*                      PHI   = (2.0*PI) - XLON    FOR XLON .GT. 0.0
C*                      IF (PHI .GT. PI) PHI = PHI - (2.0*PI)
C*
C*                   FOR CONVERSION OF (THETA,PHI) TO (LAT,LON)...
C*
C*                      XLAT  = (PI/2.0) - THETA
C*                      XLON  = -PHI               FOR PHI .LT. PI
C*                      XLON  = (2.0*PI) - PHI     FOR PHI .GE. PI
C*
C*                         FOR ABS(PHI) .LT. 2.0*PI
C*
C*                   NOTE THAT SOUTH LATITUDES AND EAST LONGITUDES ARE
C*                   HANDLED AS NEGATIVE NUMBERS AND MUST BE SO SPECI-
C*                   FIED ON INPUT/OUTPUT FROM THE SUBROUTINE.  LATI-
C*                   TUDES AND LONGITUDES ARE IN DEGREES, WHILE THETA
C*                   AND PHI ARE IN RADIANS.  THE AZIMUTH ANGLE PHI IS
C*                   THE ANGLE COUNTERCLOCKWISE FROM GREENWICH.  THE
C*                   ABSOLUTE VALUE RANGES FROM 0.0 TO 2.0*PI RADIANS.
C*                   NEGATIVE VALUES INDICATE CLOCKWISE.  THE COLATI-
C*                   TUDE ANGLE THETA IS THE ANGLE SOUTHWARD FROM THE
C*                   NORTH POLE AND RANGES FROM 0.0 TO PI RADIANS.
C*
C*   INPUT &         ICONV    = FLAG CONTROLLING WHETHER CONVERSION IS
C*   OUTPUT-                    FROM (LAT,LON) TO (THETA,PHI) (ICONV=1)
C*                              OR FROM (THETA,PHI) TO (LAT,LON)
C*                              (ICONV=2).  WHEN RETURNED AS OUTPUT,
C*                              ICONV = 1 OR 2 FOR NORMAL TERMINA-
C*                              TION OR A NUMBER GREATER THAN OR
C*                              OR EQUAL TO 10 FOR ABNORMAL TERMINA-
C*                              TION.  ABNORMAL TERMINATION CODES
C*                              FOR ICONV ARE...
C*
C*                                 10-  ERRONEOUS ICONV SPECIFIED
C*                                 11-  OUT-OF-BOUNDS XLAT
C*                                 12-  OUT-OF-BOUNDS XLON
C*                                 13-  OUT-OF-BOUNDS THETA
C*                                 14-  OUT-OF-BOUNDS PHI
C*
C*                   XLAT     = LATITUDE IN DEGREES, 0.0 TO 90.0,
C*                              POSITIVE FOR NORTH LATITUDES AND NEGA-
C*                              TIVE FOR SOUTH LATITUDES.
C*
C*                   XLON     = LONGITUDE IN DEGREES, 0.0 TO 180.0, POSI-
C*                              TIVE FOR WEST LONGITUDES AND NEGATIVE FOR
```

```
C*                                  EAST LONGITUDES.
C*
C*                   THETA    = COLATITUDE ANGLE IN RADIANS, 0.0 TO PI
C*                              RADIANS SOUTHWARD FROM NORTH, POSITIVE.
C*
C*                   PHI      = AZIMUTH ANGLE IN RADIANS, 0.0 TO PI
C*                              RADIANS, COUNTERCLOCKWISE FROM GREENWICH
C*                              LOOKING DOWN AT THE NORTH POLE.  NEGATIVE
C*                              VALUES INDICATE CLOCKWISE.
C*
C*    SYSTEM SUB-
C*    PROGRAMS
C*    USED-          NONE
C*
C*    USER SUB-
C*    PROGRAMS
C*    USED-          NONE
C*
C*    ESTIMATED
C*    CPU TIME-      TO BE SUPPLIED.
C*
C*    STORAGE
C*    REQUIREMENTS-  PROCEDURE PLUS DATA OCCUPY 154 WORDS OF
C*                   CORE STORAGE.
C*
C*    PROGRAM
C*    UPDATES-       NONE
C*
C*********************************************************************
C
      DATA TWOPI/6.2831853/,     PI/3.1415927/,  PID2/1.5707963/,
     &        DTOR/0.01745329/,  RTOD/57.295780/
C
C     DEPENDING ON THE VALUE OF THE ICONV FLAG, CONVERT (LAT,LON)
C      TO (THETA,PHI) (ICONV=1) OR CONVERT (THETA,PHI) TO
C      (LAT,LON) (ICONV=2).  FOR INPUT ICONV OTHER THAN 1 OR 2,
C      AN ERROR CONDITION (ICONV=10) IS RETURNED.
C
      IF (ICONV .EQ. 1 .OR. ICONV .EQ. 2) GO TO 100
      ICONV = 10
      GO TO 1000
  100 GO TO (200, 400), ICONV
C
C     CONVERT (LAT,LON) TO (THETA,PHI).  RETURN ERROR CONDITION
C      ICONV=11 FOR OUT-OF-BOUNDS XLAT AND ICONV=12 FOR OUT-OF-
C      BOUNDS XLON.
C
  200 IF (XLAT .LE. 90.0 .AND. XLAT .GE. -90.0) GO TO 220
      ICONV = 11
      GO TO 1000
  220 IF (XLON .LE. 180.0 .AND. XLON .GE. -180.0) GO TO 250
      ICONV = 12
      GO TO 1000
  250 THETA = PID2 - (XLAT * DTOR)
      IF (XLON) 260,260,275
  260 PHI = -(XLON * DTOR)
      IF (PHI .GT. PI) PHI = PHI - TWOPI
      GO TO 800
  275 PHI = TWOPI - (XLON * DTOR)
      IF (PHI .GT. PI) PHI = PHI - TWOPI
      GO TO 800
C
C     CONVERT (THETA,PHI) TO (LAT,LON).  RETURN ERROR CONDITION
C      ICONV=13 FOR OUT-OF-BOUNDS THETA AND ICONV=14 FOR OUT-OF-
C      BOUNDS PHI.  ADJUST COMPUTED LONGITUDE FOR DATELINE FOLD.
C
  400 IF (ABS(THETA) .LE. PI) GO TO 420
      ICONV = 13
```

```
      GO TO 1000
  420 IF (ABS(PHI) .LE. TWOPI) GO TO 450
      ICONV = 14
      GO TO 1000
  450 XLAT = RTOD * (PID2 - THETA)
      IF (PHI .LT. PI) GO TO 475
      XLON = RTOD * (TWOPI - PHI)
      GO TO 500
  475 XLON = -(PHI * RTOD)
  500 IF (XLON .GT.  180.0)  XLON = XLON - 360.0
      IF (XLON .LT. -180.0)  XLON = XLON + 360.0
C
C     NORMAL TERMINATION.
C
  800 RETURN
C
C     ABNORMAL TERMINATION.
C
 1000 XLAT  = 0.0
      XLON  = 0.0
      THETA = 0.0
      PHI   = 0.0
      RETURN
      END
```

```
CDISTAN GRT CIRCLE DIST/R. C. WHITON/26 OCT 1978
C
      SUBROUTINE DISTAN (FRMLAT, FRMLNG, TOLAT, TOLNG, GCD)
C
C**********************************************************************
C*
C*    PROGRAM ID-       DISTAN
C*    MET ANALYST-      MAJ ROGER C. WHITON, USAFETAC/DNS, EXT 5412
C*    SYS ANALYST-      MAJ ROGER C. WHITON, USAFETAC/DNS, EXT 5412
C*    PROGRAMMER-       MAJ ROGER C. WHITON, USAFETAC/DNS, EXT 5412
C*
C*    CREATED ON-       26 OCT 1978           PROJECT-  192301
C*
C*    DESCRIPTION-      THIS SUBROUTINE SUBPROGRAM COMPUTES GREAT CIRCLE
C*                      DISTANCE OVER SPHERE OF THE EARTH BETWEEN TWO
C*                      POINTS 'A' AND 'B' WHOSE (LATITUDE,LONGITUDE) IS
C*                      SPECIFIED IN INPUT ARGUMENTS.
C*
C*    METHOD-           THE CONVENTIONAL FORMULA FOR THE GREAT CIRCLE ARC
C*                      GCD IN RADIANS IS USED (REF  2), NAMELY,
C*
C*                        GCD(RADIANS) = ARCOS(SIN(FRMLAT)*SIN(TOLAT) +
C*                          COS(FRMLAT)*COS(TOLAT)*COS(TOLNG-FRMLNG))
C*
C*                      WHERE INDEPENDENT VARIABLES ARE AS DESCRIBED IN
C*                      THE INPUT ARGUMENTS BELOW.  LATER, GCD IN RADIANS
C*                      OF ARC IS CONVERTED TO GCD IN NAUTICAL MILES (NM)
C*                      OF CIRCULAR DISTANCE BY MULTIPLYING BY THE
C*                      RADIUS OF THE EARTH, 3440 NM.  GCD IS RETURNED TO
C*                      THE MAIN PROGRAM IN NM.
C*
C*    LIMITATIONS-      1.  (LAT,LON) OF DESTINATION MUST NOT BE SAME
C*                          AS (LAT,LON) OF ORIGIN.
C*
C*    REFERENCES-       1.  CONRAD AND POLLACK, 1950-  METHODS IN CLIMA-
C*                          TOLOGY
C*                      2.  HEWLETT-PACKARD, 1975-  HP-65 NAVIGATION PAC 1
C*
C*    INPUT-            FRMLAT  = LATITUDE OF POINT 'A' (ORIGIN) IN DEG
C*                      FRMLNG  = LONGITUDE OF POINT 'A' (ORIGIN) IN DEG
C*                      TOLAT   = LATITUDE OF POINT 'B' (DESTINATION) IN
C*                                DEG
C*                      TOLNG   = LONGITUDE OF POINT 'B' (DESTINATION) IN
C*                                DEG
C*
C*                      NOTE-  ALL SOUTH LATITUDES AND EAST LONGITUDES MUST
C*                      BE SUPPLIED AS NEGATIVE NUMBERS.
C*
C*    OUTPUT-           GCD     = GREAT CIRCLE DISTANCE IN NM OF ARC OVER
C*                                THE SURFACE OF THE EARTH
C*
C*    SYSTEM SUB-
C*    PROGRAMS
C*    USED-             SIN, COS, ARCOS
C*
C*    USER SUB-
C*    PROGRAMS
C*    USED-             NONE
C*
C*    ESTIMATED
C*    CPU TIME-         ON  A DEC-10 GENERAL PURPOSE COMPUTER, THIS SUB-
C*                      ROUTINE REQUIRES 2.3 CPU MILLISECONDS PER CALL.
C*
C*    STORAGE
C*    REQUIREMENTS-     PROCEDURE PLUS DATA OCCUPY 94 WORDS OF
C*                      STORAGE.
```

```
C*
C*   PROGRAM
C*   UPDATES-         NONE
C*
C**********************************************************************
C
C      CONSTANTS.  DTOR CONVERTS DEGREES TO RADIANS.  A IS THE RADIUS OF
C       THE EARTH IN NAUTICAL MILES (NM).
C
       DATA DTOR/0.01745329/, A/3440.0/
C
C      COMPUTE GCD IN RADIANS OF ARC.  MULTIPLICATION BY DTOR CONVERTS
C       DEGREES TO RADIANS.
C
       SIN1 = SIN(FRMLAT * DTOR)
       SIN2 = SIN(TOLAT  * DTOR)
       COS1 = COS(FRMLAT * DTOR)
       COS2 = COS(TOLAT  * DTOR)
       COS3 = COS( (TOLNG * DTOR)  -  (FRMLNG * DTOR) )
       GCR  = SIN1 * SIN2  +  COS1 * COS2 * COS3
       GCD  = ARCOS(GCR)
C
C      CONVERT GCD FROM RADIANS OF ARC TO CIRCULAR DISTANCE BY MULTIPLY-
C       ING BY THE RADIUS OF THE EARTH A, 3440 NM.
C
       GCD  = GCD * A
C
C      TERMINATION.
C
       RETURN
       END
```

```
CHDG    INITIAL HEADING/P.L. HEROD/12 FEB 1979
C
      FUNCTION HDG (XLAT1, XLNG1, XLAT2, XLNG2)
C
C*******************************************************************
C*
C*   PROGRAM ID-     HDG
C*   MET ANALYST-    CAPT PATRICK L. HEROD, USAFETAC/DNS, EXT 5412
C*   SYS ANALYST-    MAJ ROGER C. WHITON, USAFETAC/DNS, EXT 5412
C*   PROGRAMMER-     CAPT PATRICK L. HEROD, USAFETAC/DNS, EXT 5412
C*
C*   CREATED ON-     12 FEB 1979          PROJECT-   192301
C*
C*   DESCRIPTION-    THIS FUNCTION SUBPROGRAM FINDS THE INITIAL HEADING
C*                   (HDG) ALONG A GREAT CIRCLE COURSE FLOWN FROM POINT
C*                   'A' (XLAT1,XLNG1) TO POINT 'B' (XLAT2,XLNG2).  IT
C*                   WAS DEVELOPED FROM THE HEWLETT-PACKARD 65 (HP-65)
C*                   AVIATION PAC 1, GREAT CIRCLE NAVIGATION, APRIL
C*                   1975.
C*
C*   METHOD-         THIS FUNCTION SUBPROGRAM FIRST USES INPUT LATITUDES
C*                   AND LONGITUDES FOR POINTS 'A' AND 'B' AND CONVERTS
C*                   THEM FROM DEGREES TO RADIANS.  IT THEN USES THE
C*                   HEADING FORMULA FOUND ON PAGE 54 OF THE HP-65 AVIA-
C*                   TION PAC 1 FOR CALCULATION OF THE INITIAL HEADING
C*                   FROM POINT 'A.'  SINCE THE GREAT CIRCLE DISTANCE
C*                   (GCD) IS IN THIS HEADING FORMULA, THE FUNCTION SUB-
C*                   PROGRAM MAKES A CALL TO SUBROUTINE DISTAN, WHICH
C*                   RETURNS THE GCD IN NAUTICAL MILES (NM) FROM POINT
C*                   'A' TO POINT 'B.'  FUNCTION HDG CONVERTS GCD IN
C*                   NM TO GCD IN RADIANS. THE GCD IN RADIANS IS USED IN
C*                   THE HDG FORMULA.   HDG IS COMPUTED IN RADIANS.
C*                   THERE IS A FINAL CONVERSION PROCESS WHICH ENABLES
C*                   THIS FUNCTION TO OUTPUT THE INITIAL HEADING IN
C*                   DEGREES.
C*
C*   LIMITATIONS-    TRUNCATION AND ROUNDOFF ERRORS OCCUR WHEN POINT 'A'
C*                   AND POINT 'B' ARE VERY CLOSE TOGETHER (1 MILE OR
C*                   LESS).  INPUT DATA IS IN DECIMAL DEGREES, NOT DE-
C*                   GREES, MINUTES AND SECONDS.  NORTH LATITUDES AND
C*                   WEST LONGITUDES ARE POSITIVE NUMBERS.  SOUTH LATI-
C*                   TUDES AND EAST LONGITUDES ARE NEGATIVE NUMBERS.
C*
C*   REFERENCE-      HEWLETT-PACKARD, 1975-  AVIATION PAC 1,
C*                   GREAT CIRCLE NAVIGATION
C*
C*   INPUT-          XLAT1  = LATITUDE OF POINT 'A' (ORIGIN) IN DECIMAL
C*                            DEGREES
C*                   XLNG1  = LONGITUDE OF POINT 'A' (ORIGIN) IN DECIMAL
C*                            DEGREES
C*                   XLAT2  = LATITUDE OF POINT 'B' (DESTINATION) IN
C*                            DECIMAL DEGREES
C*                   XLNG2  = LONGITUDE OF POINT 'B' (DESTINATION) IN
C*                            DECIMAL DEGREES
C*
C*   OUTPUT-         HDG    = INITIAL COURSE HEADING ALONG GREAT CIRCLE
C*                            FROM POINT 'A' TO POINT 'B' IN DECIMAL
C*                            DEGREES TOWARD WHICH
C*
C*   SYSTEM SUB-
C*   PROGRAMS
C*   USED-           SIN, COS, ARCOS
C*
C*   USER SUB-
C*   PROGRAMS
C*   USED-           DISTAN
C*
C*   ESTIMATED
```

```
C*    CPU TIME-        TO BE SUPPLIED.
C*
C*    STORAGE
C*    REQUIREMENTS-    PROCEDURE AND DATA OCCUPY 108 WORDS OF CORE
C*                     STORAGE.
C*
C*    PROGRAM
C*    UPDATES-         NONE
C*
C**********************************************************************
C
      DATA A/3440.0/, PI/3.1415927/, DTOR/0.01745329/
C
C     CONVERT LATITUDE AND LONGITUDE FROM DEGREES TO RADIANS.
C
      XLAT1R = XLAT1 * DTOR
      XLNG1R = XLNG1 * DTOR
      XLAT2R = XLAT2 * DTOR
      XLNG2R = XLNG2 * DTOR
C
C     CALCULATE GCD IN NAUTICAL MILES (GCDNM) FROM (XLAT1,XLNG1) TO
C      (XLAT2,XLNG2).
C
      CALL DISTAN (XLAT1, XLNG1, XLAT2, XLNG2, GCDNM)
C
C     CONVERT GCD IN NAUTICAL MILES (GCDNM) TO GCD IN RADIANS (GCDR).
C
      GCDR = GCDNM / A
C
C     CALCULATE HEADING IN RADIANS (HDGR) FROM HEWLETT-PACKARD 65
C
      HDGR = ARCOS((SIN(XLAT2R) - (COS(GCDR) * SIN(XLAT1R))) /
     &        (SIN(GCDR) * COS(XLAT1R)))
      IF (SIN((XLNG1 - XLNG2) * DTOR) .LT. 0.0) HDGR = 2.0 * PI - HDGR
C
C     CONVERT HEADING IN RADIANS (HDGR) TO HEADING IN NAUTICAL MILES
C      (HDG).
C
      HDG = HDGR / DTOR
C
C     NORMAL TERMINATION.
C
      RETURN
      END
```

```
CBRLNG  BRACKET LONGITUDE/P. L. HEROD/17 JAN 1979
C
      SUBROUTINE BRLNG (XLNG, XLNGLO, XLNGHI)
C
C*****************************************************************
C*
C*   PROGRAM ID-     BRLNG
C*   MET ANALYST-    CAPT PATRICK L. HEROD, USAFETAC/DNS, EXT 5412
C*   SYS ANALYST-    MAJ ROGER C. WHITON,   USAFETAC/DNS, EXT 5412
C*   PROGRAMMER-     CAPT PATRICK L. HEROD, USAFETAC/DNS, EXT 5412
C*
C*   CREATED ON-     17 JAN 1979             PROJECT 192301
C*
C*   DESCRIPTION-    THIS SUBROUTINE SUBPROGRAM FINDS THE LONGITUDINAL
C*                   VALUES THAT BRACKET A GIVEN LONGITUDE.  THE LON-
C*                   GITUDINAL BRACKET VALUES ARE IN INCREMENTS OF 30.0
C*                   I.E., A GIVEN LONGITUDE OF 132.35 HAS BRACKET VALUES
C*                   OF 120.0 AND 150.0.
C*
C*   METHOD-         THE METHOD USED TO COMPUTE THE BRACKET LONGITUDE
C*                   VALUES IS A SIMPLE ALGEBRAIC ALGORITHM INVOKING
C*                   INTEGER ALGEBRA.  SEPARATE EQUATIONS ARE REQUIRED
C*                   DEPENDING ON WHETHER WEST (POS) OR EAST (NEG)
C*                   LONGITUDES ARE USED.
C*
C*   LIMITATIONS-    ANY LONGITUDE VALUE FROM 0.00 TO 179.99 CAN BE
C*                   USED AS INPUT BUT THE BRACKET LONGITUDE VALUES
C*                   WILL ALWAYS BE 30.0 TIMES SOME INTEGER NUMBER
C*                   (0.0, 30.0, 60.0, ETC.).
C*
C*                   IN THE CASE WHERE THE INPUT VALUE IS A BRACKET
C*                   LONGITUDE AND A POSITIVE NUMBER, THE RETURNED
C*                   LONGITUDE BRACKET VALUES WILL BE THE INPUT
C*                   VALUE PLUS 30.0.  IF THE INPUT VALUE IS A BRACKET
C*                   VALUE AND A NEGATIVE NUMBER, THE RETURNED LONGITUDE
C*                   BRACKET VALUES WILL BE THE INPUT VALUE AND THE
C*                   INPUT VALUE MINUS 30.0.  IF PLUS/MINUS 180.0 IS THE
C*                   INPUT VALUE, THE RETURNED LONGITUDE BRACKET VALUES
C*                   ARE FICTITIOUS (PLUS/MINUS 180.0 AND PLUS/MINUS
C*                   210.0).
C*
C*   INPUT-          XLNG = LONGITUDE FOR WHICH BRACKETING LONGITUDE
C*                          VALUES ARE DESIRED.
C*
C*   OUTPUT-         XLNGLO = LOWER BRACKETING LONGITUDE VALUE
C*
C*                   XLNGHI = UPPER BRACKETING LONGITUDE VALUE
C*
C*   SYSTEM SUB-
C*   PROGRAMS
C*   USED-           IFIX
C*
C*   USER SUB-
C*   PROGRAMS
C*   USED-           NONE
C*
C*   ESTIMATED
C*   CPU TIME-       TO BE SUPPLIED
C*
C*   STORAGE
C*   REQUIREMENTS-   PROCEDURE PLUS DATA OCCUPY 60 WORDS OF STORAGE.
C*
C*   PROGRAM
C*   UPDATES-        NONE
C*
C*****************************************************************
C
      DATA DELTLO/30.0/
```

```
      C
      C     BRANCH TO POSITIVE OR NEGATIVE LONGITUDE COMPUTATION.
      C
            IF (XLNG) 110,105,105
      C
      C     LOWER LONGITUDE LIMIT FOR POSITIVE LONGITUDE.
      C
        105 XLNGLO = IFIX (XLNG/DELTLO) * DELTLO
      C
      C     UPPER LONGITUDE LIMIT FOR POSITIVE LONGITUDE.
      C
            XLNGHI = XLNGLO + DELTLO
            GO TO 120
      C
      C     LOWER LONGITUDE LIMIT FOR NEGATIVE LONGITUDE.
      C
        110 XLNGLO = (IFIX(XLNG/DELTLO) - 1) * DELTLO
      C
      C     UPPER LONGITUDE LIMIT FOR NEGATIVE LONGITUDE.
      C
            XLNGHI = XLNGLO + DELTLO
      C
      C     TERMINATION.
      C
        120 RETURN
            END
```

```
CBRLAT  BRACKET LATITUDE/P. L. HEROD/02 JAN 1979
C
      SUBROUTINE BRLAT (XLAT, XLATLO, XLATHI)
C
C***********************************************************************
C*
C*   PROGRAM ID-    BRLAT
C*   MET ANALYST-   CAPT PATRICK L. HEROD, USAFETAC/DNS, EXT 5412
C*   SYS ANALYST-   MAJ ROGER C. WHITON,   USAFETAC/DNS, EXT 5412
C*   PROGRAMMER-    CAPT PATRICK L. HEROD, USAFETAC/DNS, EXT 5412
C*
C*   CREATED ON-    02 JAN 1979             PROJECT 192301
C*
C*   DESCRIPTION-   THIS SUBROUTINE SUBPROGRAM FINDS THE LATITUDINAL
C*                  VALUES IN TENTHS OF DEGREES THAT BRACKET A GIVEN
C*                  LATITUDE. THE LATITUDE BRACKET VALUES ARE IN
C*                  INCREMENTS OF 15.0, I.E., A GIVEN LATITUDE OF 27.46
C*                  HAS BRACKET VALUES OF 15.0 AND 30.0.
C*
C*   METHOD-        THE METHOD USED TO COMPUTE THE BRACKET LATITUDE
C*                  VALUES IS A SIMPLE ALGEBRAIC ALGORITHM INVOKING
C*                  INTEGER ALGEBRA. SEPARATE EQUATIONS ARE REQUIRED
C*                  DEPENDING ON WHETHER NORTH (POS) OR SOUTH (NEG)
C*                  LATITUDES ARE USED.
C*
C*   LIMITATIONS-   ANY LATITUDE VALUE FROM 0.00 TO PLUS/MINUS 89.99
C*                  CAN BE USED AS INPUT BUT THE BRACKET LATITUDE
C*                  VALUES WILL BE 15.0 TIMES SOME INTEGER NUMBER
C*                  (0.0, 15.0, 30.0, 45.0, ETC.). IN THE CASE WHERE
C*                  THE INPUT VALUE IS A BRACKET VALUE AND A POS-
C*                  ITIVE NUMBER, THE RETURNED LATITUDE BRACKET
C*                  VALUES WILL BE THE INPUT VALUE AND THE INPUT
C*                  VALUE PLUS 15.0.  IF THE INPUT VALUE IS A BRACKET
C*                  VALUE AND A NEGATIVE NUMBER, THE RETURNED LATITUDE
C*                  BRACKET VALUES WILL BE THE INPUT VALUE AND THE
C*                  INPUT VALUE MINUS 15.0.  IF PLUS/MINUS 90.0 IS THE
C*                  INPUT VALUE, THE RETURNED LATITUDE BRACKET
C*                  VALUES ARE FICTITIOUS (PLUS/MINUS 90.0 AND
C*                  PLUS/MINUS 105.0).
C*
C*   INPUT-         XLAT = LATITUDE FOR WHICH BRACKETING LATITUDE
C*                  VALUES ARE DESIRED.
C*
C*   OUTPUT-        XLATLO = LOWER BRACKETING LATITUDE VALUE
C*
C*                  XLATHI = UPPER BRACKETING LATITUDE VALUE
C*
C*   SYSTEM SUB-
C*   PROGRAMS
C*   USED-          IFIX
C*
C*   USER SUB-
C*   PROGRAMS
C*   USED-          NONE
C*
C*   ESTIMATED
C*   CPU TIME-      TO BE SUPPLIED
C*
C*   STORAGE
C*
C*   REQUIREMENTS-  PROCEDURE PLUS DATA OCCUPY 58 WORDS OF STORAGE.
C*
C*   PROGRAM
C*   UPDATES-       NONE
C*
C***********************************************************************
C
      DATA DELTLA/15.0/
```

```
C
C     BRANCH TO POSITIVE OR NEGATIVE LATITUDE COMPUTATION.
C
      IF (XLAT) 110.105,105
C
C     LOWER LATITUDE LIMIT FOR POSITIVE LATITUDE.
C
  105 XLATLO = IFIX (XLAT/DELTLA) * DELTLA
C
C     UPPER LATITUDE LIMIT FOR POSITIVE LATITUDE.
C
      XLATHI = XLATLO + DELTLA
      GO TO 120
C
C     UPPER LATITUDE LIMIT FOR NEGATIVE LATITUDE.
C
  110 XLATHI = IFIX (XLAT/DELTLA) * DELTLA
C
C     LOWER LATITUDE LIMIT FOR NEGATIVE LATITUDE.
C
      XLATLO = XLATHI - DELTLA
C
C     TERMINATION.
C
  120 RETURN
      END
```

8